TRABALHO DE GRADUAÇÃO

ESTUDO DOS EFEITOS DE IMAGENS DEGRADADAS NO PROCESSO DE RECONHECIMENTO DE OBJETOS POR REDES NEURAIS CONVOLUCIONAIS

José Gabriel Hermes Cavalcanti

Brasília, julho de 2017

ENGENHARIA
MECATRÔNICA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

ESTUDO DOS EFEITOS DE IMAGENS DEGRADADAS NO PROCESSO DE RECONHECIMENTO DE OBJETOS POR REDES NEURAIS CONVOLUCIONAIS

José Gabriel Hermes Cavalcanti

Relatório submetido como requisito parcial de obtenção de grau de Engenheiro de Controle e Automação

Banca Examinadora

Prof. Flávio de Barros Vidal, CIC/UnB _____
Orientador

Prof. Marcus Vinicius Lamar, CIC/UnB _____
Examinador interno

Prof. Cauê Zaghetto, CIC/UnB _____
Examinador interno

Brasília, julho de 2017

FICHA CATALOGRÁFICA

JOSÉ GABRIEL, HERMES CAVALCANTI
Estudo dos efeitos de imagens degradadas no processo de reconhecimento de objetos por redes neurais convolucionais

[Distrito Federal] 2017.

x, 82p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2017). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Redes neurais convolucionais 2.Imagens degradadas
3. Reconhecimento de objetos em imagens

I. Mecatrônica/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

CAVALCANTI, J. G. H., (2017). Estudo dos efeitos de imagens degradadas no processo de reconhecimento de objetos por redes neurais convolucionais. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-$n°16/2017$, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 82p.

CESSÃO DE DIREITOS

AUTOR: José Gabriel Hermes Cavalcanti

TITULO DO TRABALHO DE GRADUAÇÃO: Estudo dos efeitos de imagens degradadas no processo de reconhecimento de objetos por redes neurais convolucionais.

GRAU: Engenheiro ANO: 2017

José Gabriel Hermes Cavalcanti

Condomínio Rk, Conjunto Antares, Quadra U, Casa 27.

73252-200 Sobradinho – DF – Brasil.

Dedicatória

Dedico este trabalho a minha família, por todo apoio a mim dado, aos amigos por me permitirem descontrair em tempos difíceis, ao meu orientador por suas ideias de grande valor e a você, por qualquer que seja o motivo a que tenha entrado em contato com este texto.

José Gabriel Hermes Cavalcanti

Agradecimentos

Agradeço à Fortuna que me permitiu iniciar e seguir estudos na área de Controle e Automação.
Ao Sono, fonte de sonhos e ideias, por mais que por vezes estivesse por muito ausente.
A meus pais, que mesmo em situações conturbadas, continuaram a me apoiar nesta jornada.
Agradeço ao professor Flávio de Barros Vidal que continuou a orientar e a se dedicar a conseguir tempo para este projeto, apesar de suas próprias circunstâncias e problemas.
E também a todos os colegas, amigos e professores da Faculdade de Tecnologia ou mesmo da Universidade de Brasília como um todo, que de alguma forma contribuíram ou participaram de meu processo educacional.

José Gabriel Hermes Cavalcanti

RESUMO

Decorrente da constante evolução tecnológica e da crescente disponibilidade de dados, em especial imagens, cresce a utilização de abordagens de programação orientada a dados, em que se destacam as aplicações de aprendizagem profunda de máquinas utilizando redes neurais convolucionais. Porém a alta qualidade das imagens fornecidas para treinamento ou comumente para avaliação, contrasta com a qualidade de imagens cotidianas ou de aplicações específicas, o que leva ao questionamento dos efeitos deste tipo de situação encontrada e o seu impacto no processo de classificação realizado pela rede neural convolucional. Para contemplar e responder estes questionamentos, neste trabalho foram avaliadas cinco modelos do estado-da-arte de redes convolucionais treinadas, sujeitas à sete tipos de degradação de qualidade, mostrando a sensibilidade a este conjunto de degradações e as influências na robustez do processo de classificação de objetos em imagens.

Palavras Chave: classificação de imagens, redes neurais convolucionais, degradação de imagens

ABSTRACT

Due to constant technological evolution and the increasing availability of data, in particular images, the use of data-driven programming approaches is growing, in which stands out the applications of deep machine learning using convolutional neural networks. However, the high quality of the images provided for training or commonly for evaluation, contrasts with the quality of everyday images or of specific applications, which leads to the questioning of the effects of this kind of faced situation and its impact on the classification process performed by the convolutional neural network. In order to contemplate and answer these questions, this document evaluated five state-of-the-art models of trained convolutional networks, subject to seven types of quality degradations, showing sensitivity to this set of degradations and the influences on the robustness of the object classification process on pictures.

Keywords: image classification, convolutional neural networks, image degradations

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Símbolos Latinos

D	Profundidade de uma matriz
W	Altura de uma matriz
K	Matriz filtro ou de convolução
H	Largura de uma Matriz
M	Matriz modificada
N	Número de filtros utilizado em uma camada convolucional
P	Preenchimento de borda aplicado à uma matriz
Q	Matriz de quantização (utilizada pelo compressor JPEG)
S	Passo utilizado em uma operação de convolução
T	Matriz transformada
$\overline{W}$	Matriz de pesos

Símbolos Gregos

μ	Média de uma amostra ou matriz
σ	Desvio padrão de uma amostra ou matriz
ϕ	Função não linear

Grupos Adimensionais

i, u, x	Componentes horizontais de uma matriz
j, v, y	Componentes verticais de uma matriz
w	Elemento de uma matriz de pesos
z	Componente genérico de um vetor ou matriz

Subscritos

i, u, x	Relativo aos eixos horizontais
j, v, y	Relativo aos eixos verticais

Sobrescritos

$\bar{\cdot}$	Valor quantizado
$\hat{\cdot}$	Valor reconstruído

Siglas

CIFAR	*Canadian Institute for Advanced Research*
CNN	*Convolutional Neural Network*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
JPEG	*Joint Photographic Experts Group*
MIRC	*MInimal Recognizable Configurations*
MNIST	*Modified National Institute of Standards and Technology*
MOS	*Mean Opinion Score*
VGG	*Visual Geometry Group*
ReLU	*Rectified Linear Unit*

Capítulo 1

Introdução

1.1 Contextualização

Desde o sistema de detecção facial desenvolvido por Viola-Jones [2] aplicado nos mais diversos sistemas como câmeras digitais, passando por sistemas automáticos de checagem de nível de garrafas em uma fábrica de refrigerantes, aos sistemas visuais dos carros autônomos, nos mais diversos tópicos, a área de visão computacional tem consolidado sua presença como ferramenta capaz de substituir, ou no mínimo auxiliar, o ser humano em tarefas tediosas, complicadas ou que requerem grande e constante atenção, com capacidades e objetivos que se sobrepõem com os de aplicações da engenharia de controle e automação.

A visão computacional se divide em várias subáreas, com maior ou menor sobreposição a outros campos do conhecimento humano. Uma destas subáreas em destaque atualmente é a da utilização de algoritmos de aprendizagem de máquina para reconhecimento de objetos em imagens, em geral empregando redes neurais convolucionais profundas. Aplicações destas redes até mesmo se tornaram publicamente famosas, como nas aplicações *deepdream* desenvolvidas pelo Google [3] ou o aplicativo móvel Prisma, baseado na publicação de Gatys *et al.* [4]

Por meio de redes convolucionais em específico, problemas de classificação de imagens, cujas soluções podem ser adaptadas para uma série de problemas derivados, como rotulação ou mesmo aumento de escala de imagens, tiveram um grande salto em melhoria a partir do ano de 2012, como ilustrado pela Figura 1.1. Tal salto melhoria é comumente atribuído ao trabalho da equipe de Alex Krizhevsky [5], que venceu o desafio de classificação de imagens ILSRVC 2012, reduzindo a taxa de erros de classificação de imagens por um algoritmo computacional de 25,7% para 15,3%. A partir deste trabalho, muitos outros seguiram utilizando a arquitetura de redes neurais convolucionais, progressivamente reduzindo a taxa de erros ao patamar de 2,991% para o desafio de 2016. Para efeitos de comparação, a taxa de erros calculada pelo grupo responsável pela IMAGENET [6] para seres humanos é de 5,1%.

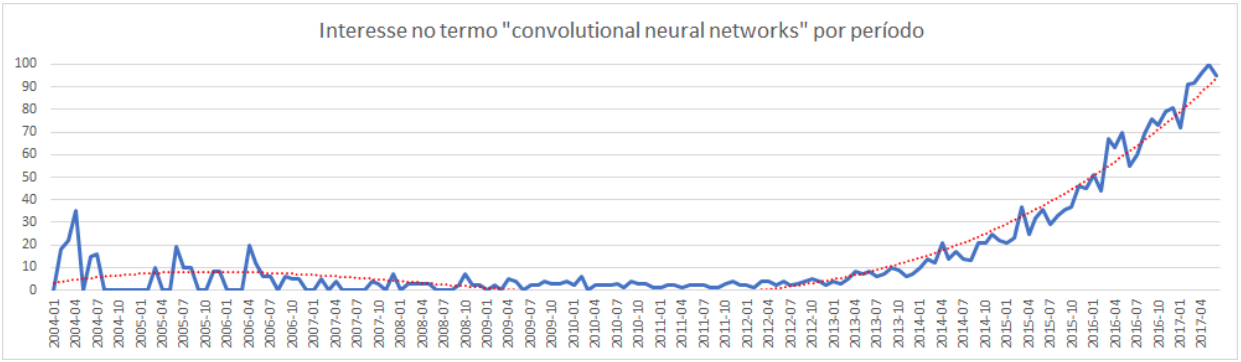


Figura 1.1: Evolução do temporal interesse no termo *convolutional neural networks* no Google [1]

1.2 Motivação

Desde os primórdios da história da computação, temas como inteligência artificial, aprendizagem de máquina, ou modelagens de bio-inspiração estiveram presentes. Em especial, um dos desenvolvimentos com grande impacto no estado-da-arte atual se deu do contínuo relacionamento destes temas, e do desenvolvimento tecnológico, com a construção de modelos neurais artificiais e posteriormente das redes neurais convolucionais.

1.2.1 Breve Histórico

A história das redes neurais começa com modelagens de mecanismos neuronais biológicos, como os trabalhos de McCulloch and Pitts [7], e a proposição de uma "lógica de limiar", cujos modelos não aprendiam autonomamente. Segue com a criação do perceptron, por Frank Rosenblatt em 1958 [8]. Todas estas abordagens permaneceram sem grandes aplicações práticas pois não havia, até o momento um algoritmo computacional que permitisse obter os pesos da rede de forma autônoma eficaz durante o treinamento, além de que tais abordagens apresentavam dificuldades em generalizar algumas funções ou operações. Estes problemas foram solucionados com a proposição do algoritmo de Retropropagação (do inglês, *Backpropagation*) creditada a Paul Werbos em sua tese de PhD em Harvard em 1974 [9] e com a utilização de camadas neurais ocultas, ambos métodos exigentes computacionalmente.

Redes convolucionais especificamente, foram propostas em 1979 por Kunihiko Fukushima [10] apesar de não utilizar o nome "rede convolucional". Foram inspiradas em pesquisas sobre o córtex visual animal, principalmente nos trabalhos de D. Hubel e T. Wiesel [11], que em publicações por volta da década de 60, escrevem sobre a ativação de neurônios ligados ao sistema visual e que tais células apresentam ativação baseada em um contexto local e dependente de características visuais específicas. Neste tipo de arquitetura a rede deixa de ser totalmente conectada, passando a considerar espacialidade a partir de núcleos dedicados a detecção de características específicas, operando como filtros, que então são transladados sobre a entrada da rede, geralmente uma imagem, gerando mapeamento destes recursos detectados por operação de convolução, que nomeia a rede.

Por não ser totalmente conectada e pelo compartilhamento de parâmetros possui menos pesos a serem determinados e passa a considerar a espacialidade dos dados, apesar de não ter a localidade como limitante.

Apesar disso, ainda não era o momento de adoção ampla das redes convolucionais, o que ocorre apenas recentemente. Um dos marcos do ínterim fora o trabalho de LeCunn [12], aplicado ao reconhecimento de imagens da base de caracteres escritos à mão MNIST.

O real marco para o início da adoção ampla de redes neurais convolucionais se dá com o trabalho de Alex Krizhevsky *et al.* [5], que em 2012 venceram o desafio anual ILSVRC de classificação de imagens utilizando redes neurais convolucionais, com uma taxa de erros de quase a metade dos valores obtidos pelo segundo colocado, que utilizava estratégias de classificadores lineares a partir da extração manual de recursos das imagens. Nos anos seguintes, estas redes seguiram sendo utilizadas pelos vencedores e demais melhores colocados no mesmo desafio, e não somente neste tipo de tarefa, já que em outras tarefas como localização de objetos em imagens ou mesmo na descrição textual de imagens, as redes neurais convolucionais tem se destacado.

1.2.2 Evolução Tecnológica

Um dos aspectos óbvios que permitiu o desenvolvimento recente das redes neurais convolucionais foi a evolução tecnológica atestada pela Lei de Moore que é discutida exaustivamente no trabalho de Patterson e Hennessy [13]. Porém, considerar apenas a evolução da quantidade de transistores utilizados não explica a adoção recente e generalizada de estratégias de aprendizagem de máquina.

Há aspectos de influência relacionados à nuances de hardware e à informação em si. Relativamente à primeira, um grande fator contribuinte ao desenvolvimento de aplicações de aprendizagem de máquina fora o desenvolvimento a partir dos anos 90 das GPUs modernas.

Tradicionalmente, a computação fora feita por meio de CPUs, que se destacam por realizar operações de multipropósito e pela facilidade em transitar por tarefas. As GPUs surgem de uma necessidade de operar com aplicações visuais de maneira intensiva, se destacando em realizar operações matemáticas utilizadas para este propósito, como multiplicações rápidas e em paralelo de vetores e matrizes, necessárias para desde a geração de realidades virtuais em tempo real de jogos digitais, incluindo até tarefas complexas de renderização em um contexto de animações por computador, em que aspectos de tempo são secundários à aspectos de carga de trabalho a executar, à medida que utilizam algoritmos exigentes computacionalmente, mas que conferem maior qualidade visual em texturas, sombreamento, reflexos, etc.

Com a progressiva adoção de GPUs por diversas indústrias relacionadas ao uso de imagens digitais, como a de jogos ou a de fotografia digital, houve a disponibilização de unidades de baixo custo de grande capacidade de processamento matemático utilizado justamente pela área de aprendizagem de máquina, em que a matemática de números reais, que pode assumir a forma de ponto flutuante, é essencial. Tal indústria de jogos, por exemplo, relaciona ao uso de GPUs por usuários domésticos, um grande mercado consumidor . Pode-se avaliar uma evolução da recente capacidade

de processamento de operações de ponto flutuante por meio da Figura 1.2, que também compara com a capacidade de CPUs. Nesta, também é possível notar a superioridade das GPUs neste tipo de processamento.

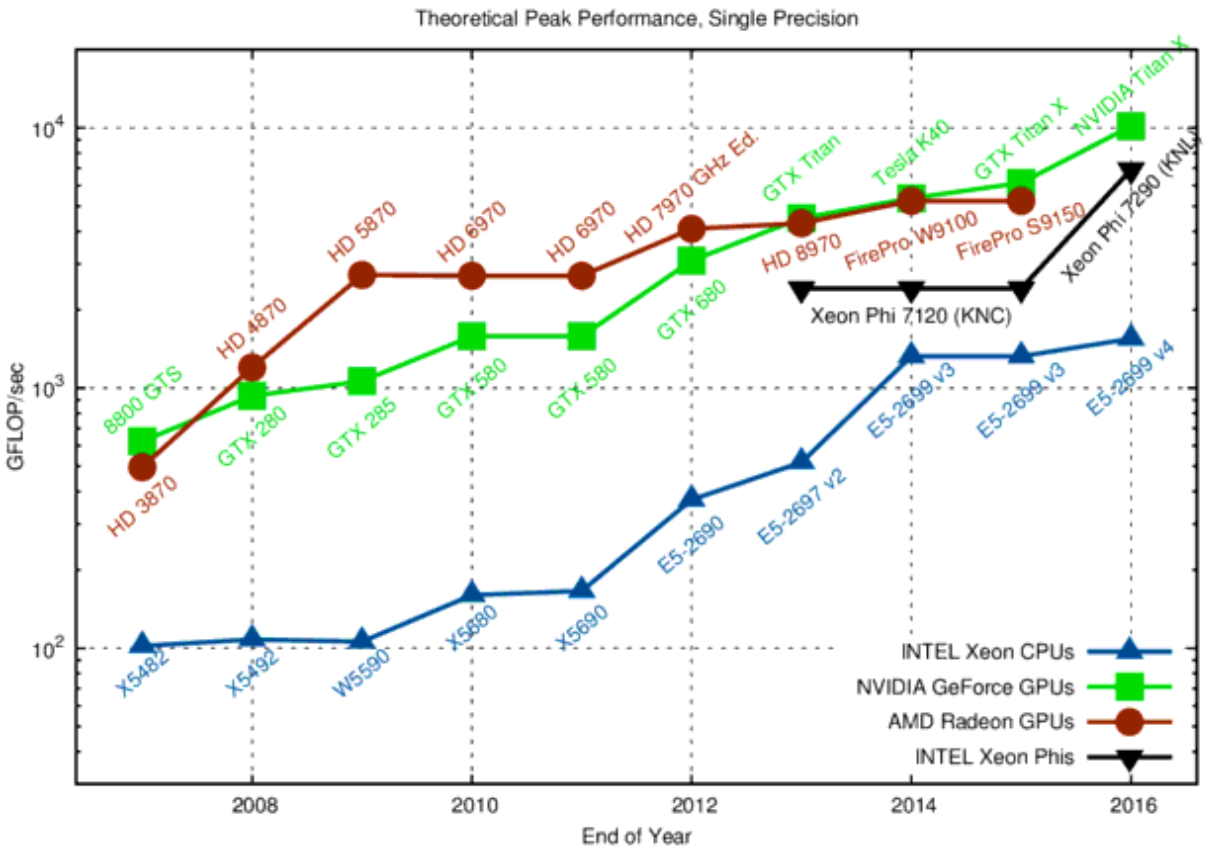


Figura 1.2: Evolução do poder de processamento de instruções de ponto flutuante pelo tempo.[14]

Relativo à informação em si, a popularização da internet gerou um grande volume de dados a serem armazenados, processados e passíveis de serem interpretados. Somente ao site Facebook, são enviadas cerca de 300 milhões de imagens diariamente [15][16]. Ao Youtube, são enviadas cercas de 400 horas de vídeo por minuto [17]. Destes dados online, uma grande quantidade fora disponibilizada publicamente, reunida e processada em grandes bases de dados construídas com o objetivo de auxiliar o treinamento e avaliação de modelos e mecanismos de aprendizagem de máquina, como por exemplo, as bases CIFAR, IMAGENET, MNIST. Tais bases se mostram essenciais nos esforços de algoritmos de aprendizagem supervisionada como os comumente empregados nas redes neurais convolucionais.

1.2.3 Justificativa

Ao panorama citado, se adicionam conceitos como o de internet das coisas, de *big data*, de ciclo de vida de produtos, e principalmente, de aspectos de qualidade de dados [18]. Cada vez mais se geram dados, porém em geral não se garante a perfeita qualidade destes. Com a necessidade de que

tais sejam processados de alguma forma, faz-se necessário não somente trabalhar os tradicionais aspectos de desempenho ou de qualidade de componentes, ou de ciclo de vida de hardware, à medida que muitas aplicações terão longos ciclos de vida, ou baixo desempenho. Mas também de como o software interage com a informação a ele alimentada, ou como a qualidade dos dados impacta na qualidade do processamento da ferramenta empregada.

Restringindo o escopo, nota-se a necessidade de analisar o impacto desta grande quantidade de informações no desempenho de redes neurais convolucionais, principalmente a influência de imagens de baixa qualidade, em outras palavras, de alguma forma, degradadas. Em especial, neste caso, como tipos específicos de degradação afetam as respostas dadas por tais redes no processo de classificação destas imagens degradadas.

1.3 Objetivo Geral

O objetivo do projeto consiste em avaliar o efeito causado por imagens artificialmente degradadas no desempenho de redes neurais convolucionais de referência em classificação de objetos, de forma a viabilizar a mensuração dos principais tipos de degradação e seus efeitos.

1.3.1 Objetivos Específicos

Dentre os objetivos específicos a serem alcançados neste trabalho, podem ser elencados os seguintes tópicos:

- Realizar estudos sobre as redes neurais convolucionais, incluindo os principais modelos treinados disponíveis na literatura atual;
- Construir uma base de dados de imagens degradadas artificialmente para a realização dos testes de acurácia e precisão dos modelos treinados de redes neurais convolucionais escolhidas;
- Definir metodologia de estudo comparativo entre os modelos escolhidos e suas eficiências em reconhecer imagens degradas a que são expostos, incluindo elaboração de métricas de avaliação de desempenho.

1.4 Divisão do Trabalho

O trabalho proposto é dividido nos seguintes capítulos: No capítulo 2 são apresentados superficialmente os principais trabalhos relacionados ao tema deste documento. O capítulo 3 introduz os conceitos básicos de degradações de imagens empregadas além de um curto desenvolvimento de redes neurais convolucionais e tópicos associados. No capítulo 4 é explicada a metodologia empregada para que fosse possível sair da delimitação de objetivos à sua realização. No capítulo 5, são expostos os resultados da metodologia construída para o estudo proposto inicialmente. Finalmente, no capítulo 6 são discutidas as conclusões deste trabalho e de como este se relaciona com trabalhos relacionados, além de propor melhorias e tópicos relacionados para estudo.

Capítulo 2

Trabalhos Relacionados

Para muitas aplicações em visão computacional, é procurado pelos seus desenvolvedores que as imagens de entrada sejam relativamente de alta qualidade. Entretanto, em certas aplicações como em sistemas de vigilância, qualidade de imagem é uma consideração importante, mas poucos fazem o seu uso, principalmente pela dificuldade de armazenamento destes tipos de imagens. Adicionalmente, com o advento de muitas aplicações de visão computacional em celulares, os requerimentos de imagens de alta qualidade podem precisar ser relaxados à medida que apesar dos constantes avanços da indústria de celulares, há aspectos de ciclo de vida de produtos antigos que produzem imagens de menor qualidade; ou que aparelhos móveis produzem imagens sem constância ou controle de ambiente, em grandes volumes ou sujeitas à grandes compactações, principalmente quando em um contexto da presença de mensageiros instantâneos.

Em aplicações de vigilância, reconhecimento de faces em imagens de baixa qualidade é uma capacidade importante para o bom funcionamento deste tipo de aplicação. Existem diversos trabalhos que tentam reconhecer faces em baixas resoluções, como no trabalho de W. Zou *et al.* [19] que estuda o reconhecimento de imagem inferiores a 16×16 pixels por meio do aprendizado de relações entre imagens pequenas e de Super Resolução (SR) e X. Ren *et al.* [20] que não utiliza métodos de super resolução. Krizhevsky *et al.* [21] apresentam bases de dados de 10 e 100 categorias (CIFAR) em baixa resolução para tarefa de classificação. Além de baixa resolução, outras distorções de qualidade de imagens podem afetar performance. Em Karam [22] é apresentado um banco de dados que considera quatro tipos de distorção de qualidade, entretanto, não avaliam a performance de quaisquer modelos nesta nova base de dados. Tao *et al.* [23] apresenta uma abordagem baseada em representações esparsas que resulta em boa performance nesta base de dados.

Para reconhecimento de caracteres manuscritos, Basu *et al.* [24] apresentam o banco de dados n-MNIST, que é uma modificação da base MNIST [25], usualmente utilizada para *benchmark*. n-MNIST adiciona ruído gaussiano, desfoque (*blur*) por movimento e redução de contraste às imagens originais. Adicionalmente, os autores propõem uma modificação das redes *deep belief* para atingir melhor acurácia nesta base de dados.

Há também o cenário de consumo de conteúdo visual, desde a internet até transmissões televisivas, em que o derradeiro examinador de aspectos de qualidade é o ser humano, seja na

escolha de um conteúdo pela qualidade, seja pela escolha de parâmetros que influenciem a qualidade deste conteúdo. A subjetividade se destaca, mas pela quantidade de conteúdo visual gerada diariamente, faz-se necessária criação de métrica objetiva de qualidade visual, que dentre outras aplicações, permitiria a maximização de aspectos de qualidade para um conjunto arbitrário de recursos. Destacam-se os trabalhos de Sheikh *et al.* [26], com a geração da base *LIVE Image Quality Assessment Database* que contém avaliações subjetivas de imagens com distorções de qualidade, e as aplicações de redes neurais para este fim por A. Bouzerdoum *et al.* [27] que utiliza um perceptron multicamadas para extrair relação entre características extraídas por programa externo à rede para prever o MOS (*Mean Opinion Score*) de imagens; e Kang *et al.* [28], que utiliza uma rede neural convolucional.

Tanto Cadieu *et al.* [29] quanto Ullman *et al.* [30] buscam comparar se os recursos e representações aprendidos pelas redes neurais artificiais são similares com os do sistema visual humano. Enquanto o trabalho de Cadieu trata de comparar representações entre mecanismos internos ao cérebro de macacos rhesus e redes neurais artificiais quando apresentadas à imagens sujeitas à variações geométricas, observando similaridade; Ullman considera performance em redes neurais profundas em recortes de baixa resolução de uma imagem, encontrando configurações mínimas de imagens (MIRC, do inglês *minimal recognizable configuration*) que são os menores recortes para os quais observadores humanos ainda podem predizer a classe correta. MIRCs são descobertos por repetidamente recortar a imagem de entrada e perguntar para observadores humanos se eles ainda podem reconhecer a imagem recortada. As regiões MIRC são borradas por que em geral representam regiões muito pequenas. Os autores testam redes profundas nas regiões MIRC e mostram que elas não alcançam performance humana.

Dodge, S. e Karam L. [31] são influenciados pelas chamadas imagens adversárias, que são imagens formadas pela aplicação intencional de perturbações de pior caso tal que resultem em uma classificação incorreta e com alta confiança por redes neurais, de acordo com o trabalho de Goodfellow *et al.* [32]. Compreendendo que as mais prováveis fontes de distorções ocorrem em processos de geração, transmissão ou armazenagem de imagens, seguem com uma investigação sobre o quanto o desempenho de redes profundas do estado da arte em classificação de imagens (CaffeNet, VGG-CNN-S, VGG16 e GoogleNet) é afetado pela utilização de degradações ordinárias como compressão JPEG, compressão JPEG2000, ruído, desfoque (*blur*) e redução de contraste, em diferentes escalas, aplicadas sobre um subconjunto da base de imagens de validação da competição ILSVRC 2012.

Em seus testes, redes se mostram sensíveis principalmente ao desfoque e ao ruído. Sugere que o desfoque elimina texturas que a rede utilizaria para efetuar a classificação e que redes mais profundas aprendem características menos sujeitas ao ruído. Ao mesmo tempo se mostram bastante resilientes a degradações advindas de compressões ou redução de contraste.

Novamente no estudo do reconhecimento facial, Kirtac *et al.* [33] avalia como as redes VGG-Face, GoogleNet e AlexNet, refinadas em bases de bancos de dados de faces, são afetadas por degradações de imagens baseadas na aplicação de ruído, desfoque e oclusões, redução de contraste e de espaço de cor. Novamente, é mencionado que as redes são muito sensíveis ao desfoque, sugerindo

que a degradação remove bordas e suaviza transições de cor, eliminando texturas utilizadas pelas redes. Também apresenta grande queda de performance para ruído. Sugere ao final que o banco de imagens de treinamento incluísse degradações como modo de buscar melhorar o resultado das redes quando validando em condições similares, ou que fossem criados modelos especializados para cada tipo de degradação.

Neste trabalho se propõe uma avaliação do desempenho de redes neurais convolucionais (CNNs) em classificar imagens de domínio diverso, expandindo o trabalho de [31], ao utilizar as redes AlexNet, CaffeNet, apesar destas duas apresentarem pequenas diferenças entre si, VGG16, VGG19 e GoogleNet. Também pretende utilizar o conjunto completo de imagens de validação da competição de 2012, além de considerar a expansão das degradações testadas, com reduções do espaço de cores e mudanças de escala.

Capítulo 3

Fundamentos Teóricos

Apesar de conceitualmente simples, para atingir os objetivos propostos é necessário conhecer a natureza das principais degradações, artificiais ou não, de imagens selecionadas para este trabalho. Bem como além de aspectos básicos de desenvolvimento e execução de redes neurais convolucionais, que em geral contém outros componentes associados, que também devem ser compreendidos para o entendimento deste trabalho.

3.1 Degradações em Imagens

Nesta seção segue um breve resumo das principais operações de degradação de imagens que foram selecionadas neste trabalho. Explicações mais detalhadas de cada uma das degradações aqui apresentadas podem ser encontradas no livro de Rafael C. Gonzalez [34], utilizado como referência principal.

3.1.1 Desfoque Gaussiano

A operação de desfocar uma imagem pode ser considerada como a aplicação de uma média ponderada entre um pixel e sua vizinhança da imagem de entrada afim de gerar um pixel de coordenada equivalente para uma imagem de saída. Resulta na atenuação das bordas ou de perda de detalhes na definição das regiões limítrofes de uma imagem.

Figura 3.1: Exemplos de imagens das bases degradadas por desfoque.

Os pesos desta média ponderada serão carregados em uma matriz que será chamado filtro,

aplicado numa operação de convolução. Os pesos do filtro terão valores proporcionais à uma distribuição gaussiana bidimensional dada pela equação. 3.1.

$$f(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y} e^{-\left[\frac{(x-\mu_x)^2}{(2\sigma_x^2)} + \frac{(y-\mu_y)^2}{(2\sigma_y^2)}\right]} \right) \tag{3.1}$$

em que σ_x é o desvio padrão aplicado no eixo horizontal de coordenadas do filtro, cuja origem se encontra no centro do mesmo. σ_y é o desvio padrão aplicado no eixo vertical de coordenadas do filtro e μ_x e μ_y são as médias das componentes horizontais e verticais do mesmo, respectivamente.

Em geral, um filtro é gerado pela aplicação direta da equação 3.1 utilizando médias nulas e desvios padrões iguais, gerando um único, que passa a ser referido como desvio padrão do filtro.

Exemplos de desfoque gaussiano podem ser visualizados na figura 3.1 que apresenta uma imagem sem desfoque e imagens construídas com filtros de tamanho 3, 5, 7, 9 e 11.

3.1.2 Redução de cores

Figura 3.2: Exemplos de imagens das bases degradadas por redução de cores.

A representação computacional de imagens depende da discretização de níveis de intensidade de cores específicas do que originalmente se trata de um espectro contínuo. Tal discretização dependerá da capacidade numérica do dispositivo de forma que o espectro seja dividido em uma quantidade finita de cores, dependendo da quantidade de bits utilizados para representar cada cor. Para 8 bits por canal de cor do sistema RGB por exemplo, há cerca de 16 milhões de cores possíveis.

Um método simples que pode ser utilizado para reduzir o número de cores de uma imagem seria, para cada um dos canais, dividir os valores de intensidade pelo máximo valor de intensidade, normalizando os canais de cor. Em seguida, multiplicar os mesmos valores pelo novo máximo valor de intensidade do espaço de cor de destino, e então, seja por arredondamento ou truncamento, tomar a parte inteira do resultado. Com as operações de divisão e multiplicação, realiza-se o mapeamento do espectro de intensidades de uma faixa de valores para outra. Com a parte inteira, garante-se que a faixa de valores de destino é discreta.

A figura 3.2 ilustra esta degradação, apresentando imagens com 256, 128, 64, 32, 16 e 8 níveis de cor por canal, o que corresponde a 8, 7, 6, 5, 4 e 3 bits por canal, respectivamente.

3.1.3 Redução de contraste

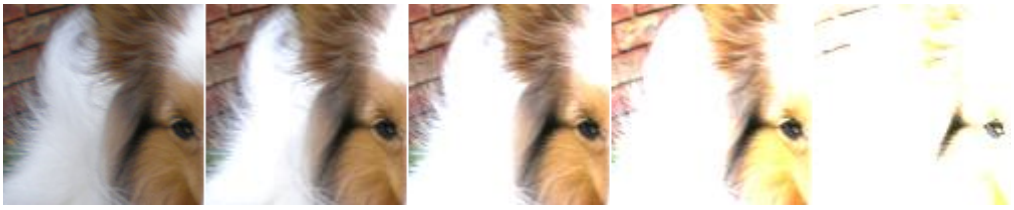


Figura 3.3: Exemplos de imagens das bases degradadas por redução de contraste.

Em geral o contraste não apresenta uma definição única de aceitação geral, podendo considerar ou não cores, mas ainda assim se define como a diferença que torna as imagens distinguíveis entre cores ou luminância, o que em uma imagem digital se traduz para a intensidade de seus pixels.

Imagens com problemas de contraste podem ser geradas devido a iluminação inadequada, seja por falta de luz suficiente, como fotografias tiradas com câmeras convencionais em ambientes escuros, ou por iluminação em excesso, como fotografias tiradas com flash muito próximo ou em ambientes já iluminados

Um método simples para diminuir o contraste de uma imagem, e análogo ao utilizado para a redução de cores disponíveis, seria efetuar a divisão de cada valor de intensidade de cada canal de cor pelo valor máximo possível, normalizando-o. Em seguida, seleciona-se um subintervalo de intensidades em que o contraste será possível, maximizando valores maiores que a parte superior e minimizando as menores que o limite inferior deste intervalo, enquanto os demais valores de intensidade serão normalizados novamente pela magnitude do intervalo. Segue com a multiplicação do máximo valor de intensidade por canal e a tomada da parte inteira do resultado.

A figura 3.3 ilustra esta degradação, apresentando imagem com contraste original, e imagens com 80%, 60%, 40% e 20% do intervalo de intensidade desta mapeados sobre a magnitude total do intervalo, conforme o procedimento explicado.

3.1.4 Ruído gaussiano

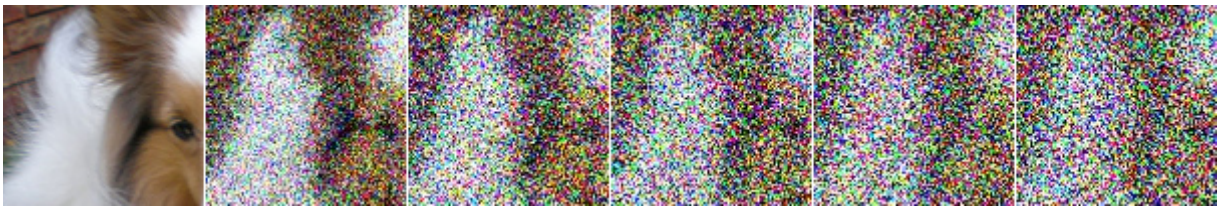


Figura 3.4: Exemplos de imagens das bases degradadas por ruído gaussiano.

Considerando a modelagem do ruído como um sinal indesejado que se soma a um sinal de referência, e que o ruído gaussiano se trata de um ruído que segue uma distribuição gaussiana, tem-se que seus valores podem ser calculados pela equação 3.2

$$p(z) = \frac{1}{2\pi\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{3.2}$$

na qual z representa intensidade, μ a média de z e σ o desvio padrão da distribuição do ruído.

Pode-se construir uma imagem ruidosa a partir de uma referência, através da equação 3.3 ,que define a transformação de uma variável aleatória que segue a distribuição normal padronizada de entrada E em uma distribuição normal de média μ e desvio padrão σ para a saída S. Começa ao gerar uma matriz de valores de ruído de mesmo tamanho espacial que a imagem de referência, que siga a distribuição gaussiana padrão, utilizando um método qualquer como por exemplo, a transformação de Box-Muller. Prossegue por transformar a matriz de ruídos padronizada em uma que siga a distribuição desejada utilizando a mesma equação 3.3. Ao somar esta matriz à imagem de referência, obtêm-se assim, uma imagem ruidosa.

$$S = \sigma \times E + \mu \tag{3.3}$$

O ruído gaussiano modela uma situação de corrupção por ruído encontrada na prática, como ruído eletromagnético ou ruídos no sensor devido à iluminação pobre ou altas temperaturas.

Exemplos de imagens degradadas por ruído gaussiano podem ser visualizados na figura 3.4, em que são apresentadas uma imagem sem degradação e imagens com ruído de desvio padrão de 0,1, 0,28, 0,46, 0,64, e 0,86.

3.1.5 Compressão JPEG

Figura 3.5: Exemplos de imagens das bases degradadas por compressão JPEG.

Um dos formatos mais populares de compressão de imagens da internet, foi criado em 1992 pelo Joint Photographic Experts Group [35], um comitê técnico formado em 1986. Especifica aspectos de codificação e testes de conformidade, apesar de carecer de software de referência. Se trata basicamente de um sistema de codificação baseado na transformada discreta de cosseno (DCT), sem delimitações quanto a modo de compressão sem perdas, formato, resolução espacial ou espaço de cores em sua especificação original.

Exemplos de compressão JPEG podem ser visualizados na figura 3.5. Nesta, são apresentadas imagens sem esta compressão, e com compressões realizadas com fator qualidade de 40, 60, 80.

De acordo com a referência utilizada [34], o algoritmo de compressão JPEG começa com uma imagem de entrada sendo primeiro dividida em blocos de tamanho 8x8, que são processados da

esquerda para a direita e de cima para baixo. Subtrai-se de seus níveis de intensidade valores 2^{k-1}, em que 2^k é o valor máximo possível, comumente $k = 8$ para 256 níveis de intensidade. Na matriz resultante, se aplica a transformada discreta de cosseno, que converte a informação do domínio espacial para o domínio da frequência, onde pode ser codificada mais eficientemente. A DCT é dada pela equação 3.4, que expressa uma transformada genérica T de um sinal bidimensional g

$$T(u,v) = \sum_{x=0}^{n-1}\sum_{y=0}^{n-1} g(x,y)r(x,y,u,v) \tag{3.4}$$

juntamente à equação 3.5, que especifica a transformada como DCT

$$r(x,y,u,v) = s(x,y,u,v) = \alpha(u)\alpha(v)cos\left[\frac{(2x+1)u\pi}{2n}\right]cos\left[\frac{(2y+1)v\pi}{2n}\right] \tag{3.5}$$

e também à equação3.6, que expressa um parâmetro da DCT.

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}}, & \text{para } u = 0 \\ \sqrt{\frac{2}{n}}, & \text{para } u = 1, 2, ..., n-1 \end{cases} \tag{3.6}$$

Então ocorre o processo de quantização, em que ocorre a compressão com perdas característica do formato. Este é dado pela equação 3.7 que utiliza uma matriz arbitrária de quantização Q, tornando a transformada T em uma transformada com perdas $\overline{T}$.

$$\overline{T}(u,v) = arredondar\left[\frac{T(u,v)}{Q(u,v)}\right] \tag{3.7}$$

A compressão é finalizada com compressões utilizando algoritmo de Huffman, explicado brevemente em [34]. Resumidamente, este algoritmo trata de ordenar um conjunto de símbolos, como por exemplo, níveis de cor de uma imagem de entrada, por suas probabilidades de ocorrência, e seguir agrupando as menores probabilidades, duas as duas, em símbolos auxiliares, até que apenas restem dois símbolos. Forma-se então uma árvore binária em que os símbolos auxiliares se ramificam sucessivamente até resultar nas folhas, que representam os símbolos originais. O próximo passo da codificação é atribuir, seguindo as ordens de probabilidades, rótulos para ramificações à esquerda e à direita de um nó raiz, de acordo com um padrão binário (por exemplo, 0 para o símbolo de maior probabilidade, 1 para o de menor). Após a atribuição completa, o código para cada símbolo original será dado pela concatenação de rótulos obtidos no caminho do nó raiz até a folha correspondente.

A descompressão de uma imagem JPEG é feita tal que para cada sub imagem, os valores codificados são decodificados pelo código de Huffman, ao que se desnormalizam os coeficientes pela equação 3.8, em que é possível perceber, quando comparando com a equação correspondente 3.7, como a quantização gera perdas.

$$\dot{T}(u,v) = \overline{T}(u,v) \times Z(u,v) \tag{3.8}$$

Prossegue-se então aplicando a DCT inversa, dada pela equação 3.9.

$$g(x,y) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} \dot{T}(u,v)s(x,y,u,v) \tag{3.9}$$

O último passo consiste em deslocar os níveis dos elementos resultantes por 2^k para obter a imagem final, que em geral será diferente da original, possivelmente se tratando de degradações, como o chamado efeito de *blocking*. Deve-se mencionar que há um modo sem perdas estabelecido posteriormente, em separado do padrão JPEG original.

3.1.6 Compressão JPEG 2000

Figura 3.6: Exemplos de imagens das bases degradadas por compressão JPEG 2000.

Outro formato criado pelo Joint Photographic Experts Group e lançado em 2000 com a intenção de substituir o formato anterior [36], o JPEG 2000 apresenta vantagens como melhorias na taxa de compressão, efeito de *blocking* não inerente ao processo de compressão, possibilidade de codificar regiões específicas por meio de "*tiles*", além de suporte à multirresolução.

Diferentemente do padrão JPEG, o JPEG 2000 é baseado em técnicas de codificação por ondaletas (*wavelets*, em inglês). Novamente, codificações são empregadas baseadas na ideia que os coeficientes de uma transformação codificam mais eficientemente que os próprios pixels. A transformação de *wavelets* gera a uma decomposição de componentes horizontais, verticais e diagonais da imagem sem a necessidade de utilizar blocos, o que reduz a propensão a efeitos de *blocking*.

O processo de codificação começa deslocando os níveis de cor individualmente, subtraindo uma quantia $2^s - 1$ de seus valores, em que s é a quantidade de bits utilizados para codificar um nível de uma cor de pixel, de maneira similar à realizada à JPEG. Segue com transformações de cor, que mapeiam os canais RGB para $YCbCr$ e podem ser no modo irreversível, como na equação de transformada 3.10, assim nomeada por, ao aplicar a transformação e sua inversa (dada por 3.11), resultar em diferenças em relação à original, por erros de precisão. Também há o modo reversível, descrito pelas equações 3.12 e 3.13. Considera-se que tais transformações melhoram a eficiência da compressão.

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,16875 & -0,3316 & 0,500 \\ 0,500 & -0,41869 & -0,08131 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.10}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1,0 & 0 & 1,402 \\ 1,0 & -0,34413 & 0,71414 \\ 1,0 & 1,772 & 0 \end{pmatrix} \times \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} \tag{3.11}$$

$$\begin{cases} Y = (R + 2G + B)/4 \\ U = R - G \\ V = B - G \end{cases} \tag{3.12}$$

$$\begin{cases} G = Y - (U + V)/4 \\ R = U + G \\ B = V + G \end{cases} \tag{3.13}$$

O algoritmo prossegue com a divisão opcional da imagem em "*tiles*", que são trechos retangulares da imagem que são processados independentemente, de modo a permitir recuperar estes trechos sem a necessidade de recuperar toda a imagem, durante a descompressão. Porém, dividir a imagem em muitos *tiles* pode produzir efeitos de bloco. Uma imagem inteira pode ser considerada um *tile*.

Segue então com a transformada discreta *wavelet* unidimensional, que pode ser vista como um par de filtros passa altas e passa baixas. É aplicada em linhas e colunas da imagem, que pode ser computada de modo reversível ou irreversível, cujos detalhes serão omitidos. A transformada produz quatro sub bandas da imagem original, em que uma se trata de uma imagem em menor resolução que a original e as demais, características de alta frequência da mesma. Esta imagem em menor escala pode ter a transformada computada novamente, e assim em diante. O padrão não delimita um número de escalas a calcular.

E então um procedimento de quantização é realizado, à medida que a quantidade de coeficientes gerados pela transformada é a mesma de elementos da imagem original, mas poucos coeficientes carregam informação relevante. Por se tratar de um procedimento também complexo, seus detalhes podem ser vistos na referência de Gonzalez [34], mas também aplica operação de arredondamento que novamente pode resultar em erros, quando o processo inverso for tomado durante a descompressão.

Degradações podem ser obtidas por cada um dos processos ditos irreversíveis.

Exemplos de compressão JPEG 2000 podem ser vistos na figura 3.6. Nesta, são apresentadas uma imagem sem esta compressão, e imagens com taxas de compressão de valor 2, 5, 10 e 15.

3.1.7 Redimensionamento espacial

Figura 3.7: Exemplos de imagens das bases degradadas por redimensionamento.

O redimensionamento, consiste em gerar uma saída de tamanho em escala relativamente a uma imagem original. Em geral, um programa de redimensionamento gera uma matriz imagem com o tamanho em escala da imagem original e então varre as localizações dos pixels da imagem de saída calculando a coordenada da entrada correspondente. Pela relação da transformação afim de redimensionamento, interpola os pixels mais próximos da coordenada de entrada para obter o valor do pixel de saída.

A mencionada interpolação é um processo de utilizar dados conhecidos para estimar valores de dados desconhecidos. A interpolação por vizinho mais próximo por exemplo, consiste em que, dado o mapeamento inverso de um pixel, o valor da saída será igual ao valor da entrada mais próxima ao mapeamento obtido, porém gera muitos artefatos visuais. Uma interpolação de melhores resultados gerais é a bi linear, que utiliza os quatro vizinhos mais próximos. Há ainda a bi cúbica, que utiliza dezesseis vizinhos mais próximos.

Apesar dos efeitos dos diferentes esquemas de interpolação serem mais marcantes durante o aumento de escala, e que neste trabalho a escala de imagens será reduzida, deve-se mencionar que, conforme será visto mais tarde, o tamanho da entrada de uma rede neural convolucional é fixo, ao que se utilizam processos de corte e redimensionamento de imagens para adequar uma imagem de entrada ao tamanho apropriado para o processamento da rede convolucional. Neste trabalho, se utiliza o esquema simples de corte único, central, de modo que aumentos de escala serão realizados sobretudo para este mesmo conjunto de imagens de tamanho espacial reduzido, 'revertendo' a redução de escala realizada. Porém, pode-se construir procedimentos em que a rede avalie toda uma série de recortes, redimensionados ou não, o que possivelmente tornaria o aumento de escala muito mais frequente e marcante.

Um exemplo da operação de redimensionamento espacial pode ser visualizado na figura 3.7, em que são demonstradas escalas original (100%), 75%, 50% e 25%.

3.2 Redes Neurais Convolucionais

Havendo um sinal de entrada matricial (como uma imagem) para o qual se deseje realizar uma classificação por meio da aplicação de uma função sobre este sinal, saindo de um espaço n×m dimensões do sinal de entrada para um espaço de k categorias de saída, é possível construir uma função de classificação linear, simplesmente consistindo do produto de uma matriz pelo sinal de

entrada ao que se soma um viés, seguindo forma da relação 3.14, em que a classe seria determinada pelo maior valor resultante no vetor de saída.

$$S = \overline{W}M + B \tag{3.14}$$

em que M é a entrada, S a saída, $\overline{W}$ uma matriz de pesos, e B um vetor de desvios (*bias*). Se a saída possui dimensão $k \times 1$ em que k representa a quantidade de classes existente, o produto das matrizes M e $\overline{W}$ deverá ter dimensões compatíveis. De modo a simplificar este produto matricial, é possível determinar arbitrariamente que para utilizar uma imagem como entrada, esta teria que ser transformada em um vetor modificado M de dimensão $(W.H) \times 1$ pela concatenação de todas as colunas de todas os canais de cor, e em que W é sua largura e H a altura originais. De modo a simplesmente obter a consistência dimensional, $\overline{W}$ deverá ter dimensão $k \times W.H$. B possui mesma dimensão de S. Tanto este vetor de desvios quanto a matriz de pesos podem ser determinados por procedimentos de aprendizagem de máquina. À esta estrutura dá-se o nome de classificador linear.

Porém, seria esperado que o uso de estruturas lineares como a proposta resultaria em resultados pobres, em um ambiente amplo, sem controle estrito de condições de entrada para o algoritmo. Considerando que para o exemplo dado anteriormente, uma linha da matriz de pesos tem a mesma dimensionalidade que uma imagem transposta, pode-se considerar a mesma como uma imagem também. Esta seria algo como um padrão para a classe que se deve classificar, com fortes limitações das características que deve reconhecer, à medida que, ao se basear apenas em uma imagem completa, este método seria incapaz de eliminar o fundo de uma imagem. Seria então esperado que se os pesos para uma classe exemplo ¨pássaro¨ forem determinados para um conjunto de imagens com pássaros voando, o método confiaria muito nas tonalidades do céu e possivelmente resultaria em falhas quando apresentando fotos de um pássaro em pouso.

Como solução para estas questões, poder-se-ia então adotar a estrutura de redes neurais, uma evolução desta modelagem de classificadores, mas que originalmente se baseou em modelagens do sistema neurológico de seres vivos. Um neurônio computacional utilizaria a equação 3.15, em que ϕ expressa uma função não linear.

$$s = \phi(\sum w_i m_i + b_i) \tag{3.15}$$

As mais populares funções não lineares utilizadas são a Sigmoid, dada pela equação 3.16,

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{3.16}$$

a tradicional Tangente Hiperbólica, expressa pela equação 3.17

$$\phi(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}} \tag{3.17}$$

e a mais recentemente popular *Rectified Linear Unit* (ReLu), dada pela equação 3.18.

$$\phi(z) = \begin{cases} 0, & \text{para } z < 0 \\ z, & \text{para } z \geq 0 \end{cases} \tag{3.18}$$

Não linearidades vezes são consideradas como estruturas em separado, aplicadas sobre a saída dos neurônios, o que em um diagrama corresponderia a serem posicionadas em série com os mesmos.

Segundo Guyton *et al.* [37], em uma rede neural biológica, cada neurônio recebe sinais por seus dendritos e envia sinais por meio de seu axônio. Tais sinais são de natureza eletroquímica, medidos como potenciais elétricos entre partes da célula. Caso o sinal que se estabelece nos dendritos seja maior que o chamado limiar excitatório, um sinal de saída é gerado, por meio do deslocamento do potencial de ação pelo corpo celular, até o axônio. Cada sinal de resposta neuronal possuirá a mesma intensidade elétrica, mas diferentes períodos de ativação, períodos estes que por acumulação de íons, contribuirão em maior ou menor grau para a propagação do sinal pelos neurônios vizinhos.

Modela-se aspectos de distâncias entre os neurônios e frequência de ativação como a aplicação de pesos, sinais algébricos simulam a natureza do sinal neural (inibitório ou excitatório). Redes neurais também possibilitam uma estruturação multicamadas, isto é, neurônios são estruturados de modo que não se comuniquem internamente, em uma mesma camada, mas conjuntamente aceitem entradas e apresentem saídas, possibilitando a formação de pilhas de camadas, o que permite estruturas de processamento mais complexas ao custo de memória e poder de processamento. Até então, tem-se a estrutura chamada "totalmente conectada", em que neurônios de uma camada superior se conectam a todos os neurônios de uma camada inferior, ponderando as saídas de cada um deles.

O desenvolvimento teórico, que se mistura com o histórico, prossegue com as pesquisas de David Hubel e Torsten Wiesel [11], que entre as décadas de 1950 e 1970 produziram importantes pesquisas sobre a percepção visual e que influenciaram também outras áreas científicas. Em especial, em trabalhos de 1962 e 1968, demonstraram que neurônios do córtex visual de cobaias animais respondiam à estímulos realizados em certas regiões do campo visual (o campo receptivo) e respondiam à padrões específicos, como linhas horizontais ou verticais.

Ao que leva ao desenvolvimento do que então será conhecido como rede convolucional, que atende à estas duas características mencionadas, do campo receptivo à resposta à padrões, ou características, específicos. Antes de adentrar no assunto, primeiramente cabe uma breve explicação sobre a operação de convolução que substituirá a multiplicação convencional de matrizes, sendo uma operação que expressa a sobreposição acumulada de uma função ou sinal por outra função, enquanto uma translada pela outra, em outras palavras, a ponderando, definida de modo mais adequado matematicamente, pela chamada integral de convolução entre dois sinais f e g (Eq. 3.19). Um exemplo de convolução entre dois sinais pode ser visualizada na Fig. 3.8.

$$(f * g)(x) = s(x) = \int_{-\infty}^{\infty} f(u)g(x-u)du \tag{3.19}$$

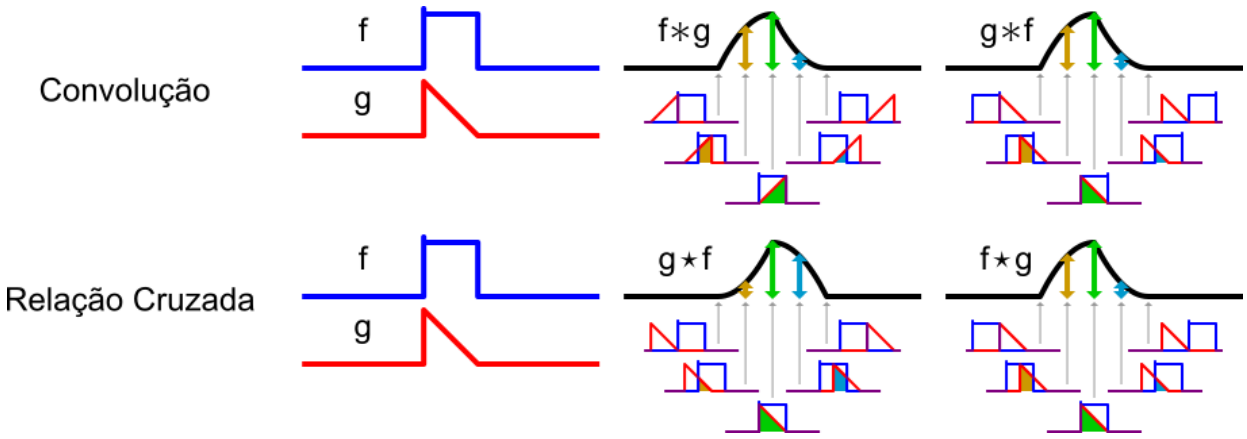


Figura 3.8: Exemplos de convolução e relação cruzada entre dois sinais [38]

Trabalhando com dados discretos, como amostragens digitais de um sensor durante um intervalo de tempo, a integral é substituída por um somatório, como na Eq. 3.20.

$$(f * g)(x) = s(x) = \sum_{u=0}^{x} f(u).g(x-u) \tag{3.20}$$

Para casos de convoluções aplicadas em apenas uma dimensão. Quando se utilizar mais de um eixo de uma vez, como no caso em que os dados são bidimensionais, como imagens, o somatório de convolução é dado pela Eq. 3.21. Em que I usualmente é chamado de entrada, ou imagem de entrada e K de *kernel*, filtro ou mesmo matriz de convolução. A saída é frequentemente chamada de mapa de características.

$$(I * K)(i,j) = S(i,j) = \sum_{u}\sum_{v} I(u,v)K(i-u, j-v) \tag{3.21}$$

Na realidade, a operação utilizada comumente nesta área de estudos não se trata de convolução, mas de correlação cruzada, em que ao invés de realizar uma subtração, aplica-se a soma, como pode ser observado, ao comparar a Eq. 3.19 com a definição da integral de correlação cruzada dada pela Eq. 3.22, em que o f^* indica a operação de complexo conjugado sobre f.

$$(f \star g)(x) = \int_{-\infty}^{\infty} f^*(u)g(u+x)du \tag{3.22}$$

Tal operação gera formatos equivalentes às equações 3.20 e 3.21, as equações 3.23 e 3.24 respectivamente.

$$(f \star g)[x] = \sum_{u=0}^{x} f^*[u].g[x+u] \tag{3.23}$$

$$(I \star K)(i,j) = S(i,j) = \sum_{u}\sum_{v} I(i+u, j+v)K(u,v) \tag{3.24}$$

Geralmente utiliza-se o termo “convolução” ou “convolucional”, apesar de incorreto, por se tratar de um termo usual, tradicional dos campos de aprendizagem de máquina ou mesmo visão computacional.

Pensando com os padrões de evolução de classificadores que se seguem neste texto, uma estrutura que utilize a convolução como base deverá ter a matriz de convolução como matriz de pesos, que, para a construção de um classificador, deverá ter seus pesos determinados algoritmicamente. Um algoritmo de aprendizagem de máquina que utiliza a relação cruzada ao invés da convolução obterá um filtro invertido em relação ao que seria obtido pelo outro. A partir deste momento, este texto seguirá com a utilização dos termos “relação cruzada” e “convolução” indistintamente.

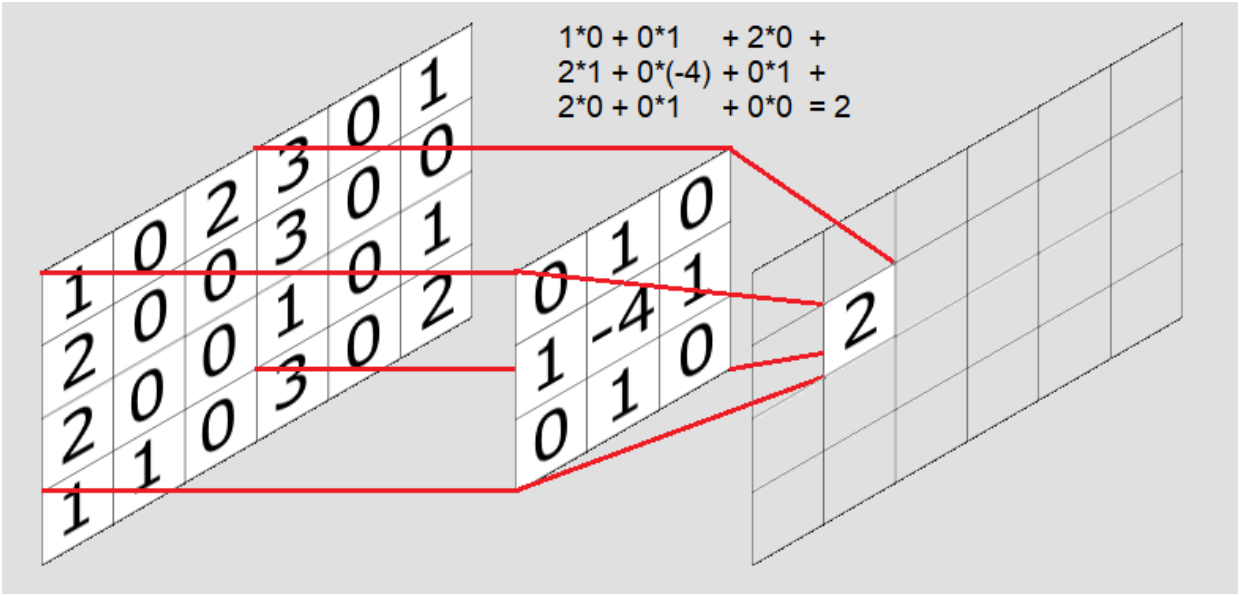
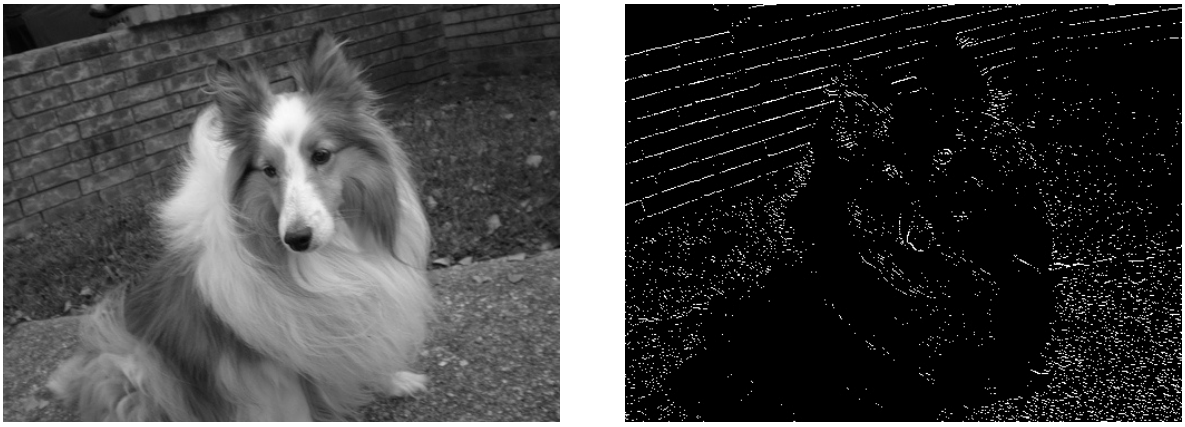


Figura 3.9: Ilustração do procedimento de convolução.

A operação de convolução equacionada anteriormente pode ser compreendida de um modo menos formal como: dado uma matriz base, como uma imagem por exemplo, e uma outra matriz a ser utilizada como filtro, realiza-se a operação de modo ilustrativo pelo deslocamento do filtro por toda a imagem de entrada, a cada uma quantidade de passos S, tanto horizontal quanto verticalmente. A cada posição tomada pelo filtro, gera-se um valor de saída, dado pela soma do produto termo a termo dos elementos sobrepostos da imagem e do filtro. A referência de coordenadas do filtro será sua posição central. A figura 3.9 ilustra o cálculo de um valor de saída para uma posição do filtro sobre uma matriz de entrada.

Um exemplo de aplicação de convolução pode ser visualizado na Figura 3.10, em que o filtro K expresso na Equação 3.25 é aplicado à imagem em escala de cinza 3.10(a) ao que então se utiliza o valor de intensidade 63 como limiar para a tornar a imagem binária, de modo a obter o resultado 3.10(b).

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.25}$$



(a) Imagem Original

(b) Resultado da Convolução

Figura 3.10: Exemplo de filtragem por convolução.

Há então neste exemplo, pelo conjunto de um operador de convolução e de limiar, um simples detector de bordas. Ao atentar que ao tornar a imagem binária aplica-se uma não linearidade, é possível então considerar este detector de bordas como uma rede neural simples, em que o cálculo matricial é determinado por convolução e cujos pesos já estão aprendidos, em sua passagem direta. Nota-se que ao aplicar o filtro à imagem de entrada é gerada uma imagem de saída contendo dados específicos sobre a original, ou em outras palavras, características de interesse. Generaliza-se isso e obtém-se o nome para a matriz resultante de mapeamento de características. De modo relacionado, ao se considerar que diferentes filtros extraem ou apresentam respostas à diferentes características da imagem de entrada, conclui-se que há um comportamento similar ao relatado para neurônios do córtex visual, como mencionado anteriormente.

Nota-se também que há a chamada equivariância à translação. Ser equivariante significa que se a entrada muda, a saída muda de maneira compatível. No caso, isso quer dizer que se a operação de convolução é aplicada à uma região da imagem, a saída será a mesma caso essa região seja deslocada pela imagem, o que gerará uma equivalência entre aplicar o deslocamento antes ou depois da convolução. No exemplo dado, procedendo cuidadosamente com aspectos de vizinhança, um recorte sobre a imagem apresentará uma mesma detecção de borda não importa a região em que este recorte esteja localizado.

Tal característica resulta em uma melhoria em relação às arquiteturas simples de neurônios completamente conectados, em que a saída depende muito de em que região da entrada se encontra uma característica à medida que não se pode garantir que os pesos que serão utilizados em uma localização serão os mesmos para outra. Se necessariamente uma rede devesse apresentar as mesmas respostas em diferentes regiões da entrada, algum tipo de condição deveria ser imposta de modo

que os pesos fossem apropriados. Em um contexto de aprendizagem de máquinas, em geral a condição seria que diversas imagens transladadas fossem fornecidas como exemplos de treinamento de modo que os pesos fossem ajustados corretamente. Redes convolucionais em contraste, resultam naturalmente em algoritmos de detecção de características mais robustos, que não necessitam ser treinados exaustivamente para diferentes posicionamentos. Entretanto, deve ser citado que estas não são equivariantes à rotação ou mudanças de escala, de tal forma que devem ser fornecidos exemplos adequados de imagens no treinamento caso deseje-se operar com rotações e diferentes escalas.

Além da equivariância translacional, há como vantagem a economia de memória utilizada, já que por exemplo, uma rede totalmente conectada com uma imagem de $32 \times 32 \times 3$ de tamanho como entrada apresentaria $32 \times 32 \times 3 = 3072$ pesos para um único neurônio na primeira camada. Uma camada convolucional, em comparação, com filtro de tamanho declarado 5×5 aplicada à mesma imagem, irá requerer $5 \times 5 \times 3 = 75$ pesos. Porém, estritamente se tratando de camadas, serão necessários $n \times 3072$ pesos para uma camada completamente conectada, em que n representa o número de neurônios (ou de saídas) da mesma, enquanto todas as saídas de uma camada convolucional serão geradas com o mesmo conjunto de pesos, pela aplicação da operação não somente em uma região, mas por toda a imagem, ao que se for considerada uma modelagem de um neurônio para cada saída, haverá o chamado compartilhamento de pesos entre neurônios. Além disso, tais neurônios serão sensíveis apenas à uma região de entrada (o campo receptivo), novamente de modo similar ao relatado sobre o funcionamento do córtex visual animal.

Em se tratando de dimensionalidade de redes convolucionais, o tamanho do filtro é equivalente ao do campo receptivo. As conexões são locais no espaço em altura e largura, mas não em profundidade, em que a conexão será completa. Não há qualquer exigência quanto à aplicação do filtro para todas as posições possíveis para uma entrada, de modo que é possível aplicá-lo intermitentemente, a um intervalo que será chamado de passo. Também há uma questão relativa ao cálculo dos valores de borda da saída, uma vez que a posição de referência da saída é correspondente à posição central do filtro, de modo que se o filtro tiver dimensões maiores do que a unitária, as bordas originais não serão consideradas de maneira central na saída, o que pode não ser o desejado. De modo a considerar estas bordas, comumente se aplica um preenchimento de borda dado como hiperparâmetro, isto é, um parâmetro dado pelo projetista da rede. Frequentemente tal preenchimento é feito com valores de intensidade zero e tamanho arbitrário, que pode ser escolhido para tanto dimensionar a rede, quanto para garantir que cada elemento da entrada será multiplicado por cada elemento do filtro ao menos uma vez.

Se há uma quantidade de filtros N, com dimensões laterais iguais de tamanho F, aplicados a passos S, com preenchimento de borda P em uma camada convolucional que aceita um volume de entrada de dimensões $W_1 \times H_1 \times D_1$, em que W_1 e H_1 são respectivamente a largura e altura, e D_1 a profundidade, o volume de saída é definido por $W_2 \times H_2 \times D_2$, em que W_2 é dado pela equação 3.26, H_2 é dado pela equação 3.27 e D_2 é definida como expresso pela equação 3.28.

$$W_2 = \frac{W_1 - F + 2P}{S} + 1 \tag{3.26}$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1 \tag{3.27}$$

$$D_2 = N \tag{3.28}$$

Com este equacionamento, é possível controlar o dimensionamento da saída. Por exemplo se for desejado que se preserve as dimensões de entrada, com passo unitário, o preenchimento de borda $P = \frac{F-1}{2}$ atinge este objetivo. Resta dizer que como o tamanho de saída se relaciona diretamente com o translado do filtro pela entrada, os resultados de 3.26, 3.27 e 3.28 devem ser todos inteiros.

Até o momento foram apresentadas estruturas básicas para a construção de redes neurais convolucionais, com camadas de neurônios comuns, chamadas de totalmente conectadas, camadas de não linearidades e camadas convolucionais. Um último conceito a descrever é o *pooling* (em português, agrupamento) que similarmente às camadas de não linearidades e diferentemente das demais, não consiste de uma função de aplicação de pesos e produtos sobre a entrada. Conforme seu nome traduzido, se trata de agrupar, utilizando alguma função, elementos de uma vizinhança da entrada, geralmente resultando em uma saída de tamanho inferior. As funções mais comuns utilizadas neste agrupamento são a média, ema saída é a média das entradas ou a operação de máximo valor. Similarmente às equações 3.26, 3.27 e 3.28, de camadas convolucionais, tem-se que para uma camada de *pooling* com entrada de volume $W_1 \times H_1 \times D_1$, o volume de saída $W_2 \times H_2 \times D_2$ é determinado pelas equações 3.29, 3.27 e 3.31.

$$W_2 = \frac{W_1 - F}{S} + 1 \tag{3.29}$$

$$H_2 = \frac{H_1 - F}{S} + 1 \tag{3.30}$$

$$D_2 = D_1 \tag{3.31}$$

Se for considerado que as camadas convolucionais reconhecem características de imagens, ao associá-las com *pooling*, obtém-se um conjunto invariante a pequenas modificações sobre a entrada. À medida que se obtém uma estatística sobre uma vizinhança, a saída apresentada pelo conjunto assume uma significação maior de "presença de característica" em contraste com uma de "localização de característica", de modo que a combinação resulta de uma menor dependência de localidades, ou em uma consequente maior capacidade de generalização.

Muitos outros conceitos, camadas, operações e características podem ser associadas à redes neurais convolucionais, gerando novas funcionalidades e potencialidades, mas que seria contraproducente mencionar neste documento. Estão presentes na referência de Goodfellow *et al.* [39] utilizada para elaboração desta seção, que explora e explica diversas estruturas presentes na área de aprendizagem profunda de máquinas.

Capítulo 4

Metodologia

A metodologia desenvolvida para atingir o objetivo de avaliar o desempenho de redes neurais convolucionais em classificar imagens degradadas consistiu na execução sequencial de passos simples que progridem do objetivo à sua concretização. Em um primeiro momento, isto significou analisar as possibilidades e a literatura com a finalidade de delimitar aspectos e detalhes iniciais do trabalho, verificando tópicos como possibilidades e adequação ao tema, o que é exposto na Seção 4.1. Nesta mesma seção, foram escolhidas as arquiteturas para representar redes neurais convolucionais, que são brevemente explicadas individualmente na Seção 4.2. Prossegue-se com uma definição básica de procedimentos de execução na seção 4.3 como de geração de bases de imagens degradadas e classificação destas e então, com os resultados, na seção 4.4 elabora-se métricas que avaliem o que se deseja estudar, isto é, o quão bem as redes escolhidas se comportam quando expostas à degradações em suas entradas.

4.1 Definição da Base de Dados

Um dos aspectos de grande influência para a recente exposição e crescente uso de redes neurais convolucionais foi a geração de amplas bases de dados de imagens rotuladas, em destaque a IMAGENET [6], uma base de mais 14 milhões de imagens que segue a hierarquia WordNet, que registra conceitos chamados *synsets* apresentando uma média de cerca de 500 imagens por nó. A IMAGENET, ssociada ao conjunto de desafios anuais ILSVRC [40], se tornou referência não somente como fontes de dados, mas também como expositora de redes de crescente desempenho na tarefa de classificar imagens. Ao que é possível notar que muitas das redes classificatórias de destaque foram avaliadas na mesma categoria do desafio, que se manteve com poucas alterações, o que permite acompanhar facilmente a evolução tecnológica.

De forma que a base de dados escolhida para este trabalho fora a de validação do ILSVRC 2012, composta por 50.000 imagens, de 1.000 categorias, por possuir uma tabela-verdade divulgada, além de se tratar de base também utilizada em trabalhos relacionados. Tal utilização fora fator de influência também para a escolha das redes convolucionais adotadas, o que, entretanto, viria a depender também de aspectos de executabilidade.

Excluindo aspectos de hardware, necessitava-se efetivamente de implementações das redes escolhidas. Dentre os diversos *frameworks* disponíveis para a construção e execução de CNNs, como por exemplo, TensorFlow [41], Theano [42] ou Torch [43], fora escolhido a Caffe [44], por possibilitar a utilização de grande número de redes pré-treinadas.

Da consideração destes fatores foram escolhidas as redes AlexNet [5], CaffeNet [45], GoogleNet [46], VGG [47] de 16 camadas e VGG de 19 camadas.

Por fim, as degradações escolhidas a serem aplicadas ao banco de imagens são de ampla utilização, em geral modelam efeitos e situações factíveis no cotidiano e em vários cenários em que o uso de CNNs se julga como possível, e são listadas por: desfoque (*blur*), redução dos níveis de cores, redução do contraste, ruído gaussiano, compressão JPEG, compressão JPEG 2000 e redimensionamento espacial (redução de escala), estas apresentadas no Capítulo 3.

4.2 Redes

As redes utilizadas neste documento representam arquiteturas padrão elaboradas pelos seus desenvolvedores, treinadas com a base de dados do desafio ILSVRC (que consiste de 1,2 milhões de imagens, em 1000 categorias). Para este trabalho, foram obtidas as redes pré-treinadas disponíveis no "*model zoo*" do *framework* de aprendizagem profunda de máquinas Caffe [44].

4.2.1 AlexNet

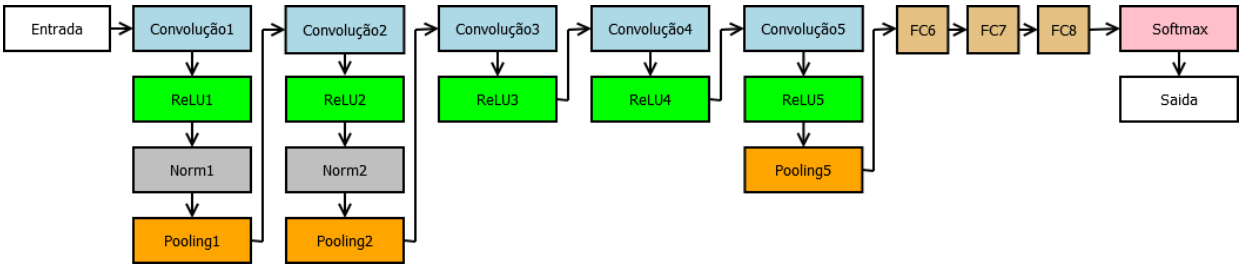


Figura 4.1: Esquema da rede AlexNet.

Trabalho responsável pela popularização recente das redes convolucionais. Sua arquitetura fora responsável pela vitória no desafio ILSVRC em 2012, com uma ampla margem de vantagem sobre o segundo colocado, apresentando como característica marcante o encadeamento de camadas, sobretudo convolucionais, formando uma arquitetura profunda. Dentre seus criadores [5] se encontra Alex Krizhevsky, cujo nome apelida a rede. Na introdução de seu documento escrito menciona como possibilitadores para a construção da rede, tanto aspectos de bases de dados amplas recentes quanto aspectos tecnológicos como a implementação de algoritmos de convolução eficientes.

A arquitetura da rede mostrada na Figura 4.1, consiste de cinco camadas convolucionais seguidas por três camadas completamente conectadas. Especificamente, para uma entrada 227 por 227, a primeira camada possui 96 filtros de dimensão 11x11x3 aplicados a um passo de 4 pixels. A

segunda possui 256 filtros de dimensão 5x5x48. A terceira possui 384 filtros de dimensão 3x3x256. A quarta, 384 filtros de dimensão 3x3x192. A quinta possui 256 filtros de tamanho 3x3x192. As camadas completamente conectadas possuem 4096 neurônios cada, totalizando cerca de 60 milhões de parâmetros.

Utiliza como não linearidades ReLUs, que além da vantagem de não demandar tanto poder computacional quando em comparação com a outra não linearidade de uso mais comum, a tangente hiperbólica, não apresenta problemas quanto à saturação por conta de sua entrada. Também utiliza *pooling* com sobreposições, ou seja, uma posição consecutiva da operação se sobrepõe à anterior. No entanto esta técnica não se provou de adoção tão geral quanto as ReLUs.

Fora utilizada a replicação do modelo disponível no *model zoo* do *framework* Caffe. Possui como diferenças ao modelo original o treinamento sem reiluminação de imagens e a inicialização de desvios (*bias*) para 0.1 ao invés de 1, pois segundo sua página da web, o peso unitário resultava em processo deficiente de aprendizagem da rede. Fora disponibilizada a iteração 360000 e resulta em uma acurácia nominal de 57.1% top1 e 80.2% top5 para o conjunto de validação, aplicando recorte central nas imagens utilizadas para a obtenção desta estatística, necessário devido à dimensão fixa da entrada.

4.2.2 CaffeNet

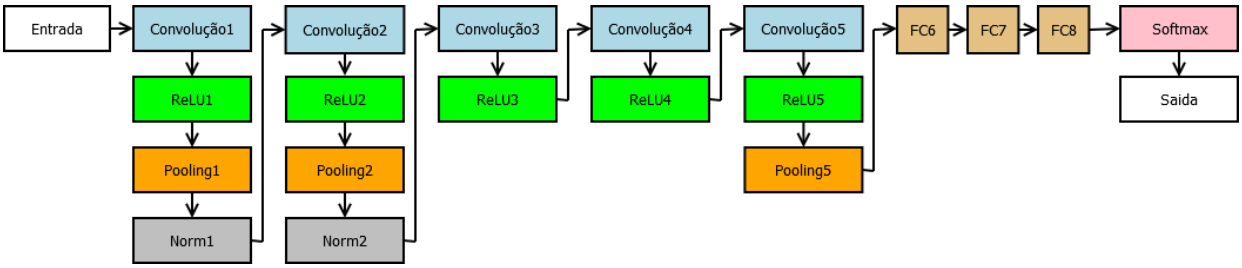


Figura 4.2: Esquema da rede CaffeNet.

Modelo utilizado (ilustrado na Figura 4.2) é uma replicação da AlexNet com a diferença de ordenamento entre as camadas de *pooling* e normalização, obtida no *model zoo* do *framework* Caffe. Fora disponibilizada na iteração 310000 com acurácia nominal de 57.4% top1 e 80.4% top5 para o conjunto de validação, aplicando recorte central nas imagens utilizadas para a obtenção desta estatística.

4.2.3 GoogleNet

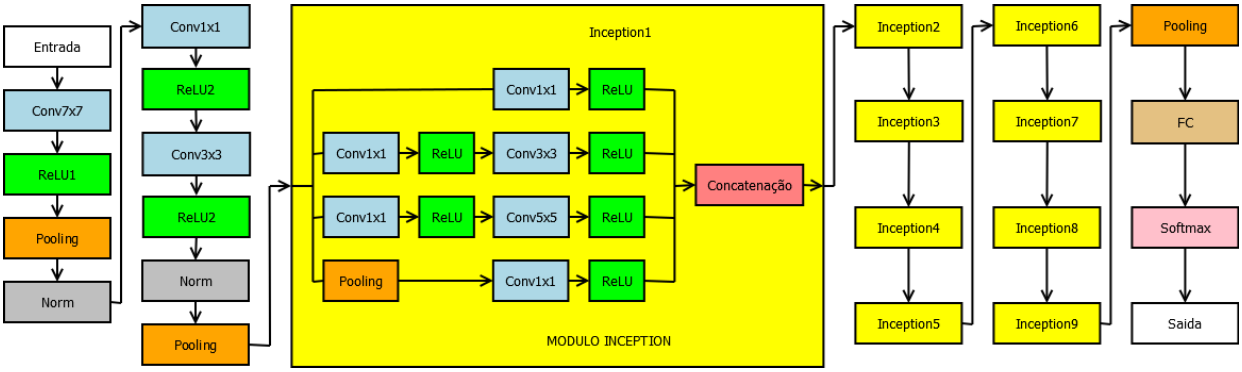


Figura 4.3: Esquema da rede GoogleNet.

Uma das arquiteturas ganhadoras do desafio ILSVRC 2014, originalmente mencionada no artigo de Szegedy *et al.* [46], motivada pela constatação de que as maneiras mais utilizadas até então de aumentar o desempenho em classificar de uma rede, que seriam aumentar ou a quantidade ou a dimensionalidade dos filtros, geram uma grande quantidade de parâmetros, que além de se relacionarem à um grande aumento da necessidade de poder computacional, resultam em uma quantidade pequena de parâmetros relevantes.

Sugere então que fossem realizadas implementações esparsas, porém ressalta que o estado da arte não é favorável à utilização de estruturas esparsas, por implementações ou algoritmos, o que não se aplica a estruturas densas. Nasce a arquitetura *Inception* como um estudo de caso de aproximação de um tipo de estrutura pelo outro, mas os autores se dizem incertos se a fundamentação na aproximação de estruturas esparsas por densas condiz realmente com os bons resultados obtidos na tarefa de classificação.

A arquitetura *Inception* sugerida se trata de uma camada em que diferentes tipos de operações são realizadas e os resultados são concatenados na saída. Esta proposta intuitivamente, permitiria que diferentes características pudessem ser disponibilizadas para as camadas superiores, aumentando o poder discriminativo da rede. O modelo principal apresenta em paralelo convoluções 1×1, convoluções 1×1 seguidas de 3×3, convoluções 1×1 seguidas por 5×5 e *pooling* 3×3 seguido de convoluções 1×1, cujas saídas são concatenadas para gerar a resposta do módulo *Inception*.

Deve-se destacar o papel de convoluções 1×1, que apesar de não operar nas dimensões laterais de uma entrada, permanece operando em canais, reduzindo esta dimensão dos dados e ajudando a diminuir a quantidade de parâmetros. Outro fator a ser considerado é que aumenta a não linearidade das redes, pois incluem ReLUs.

A arquitetura GoogleNet, que pode ser visualizada na figura 4.3, então se constrói a partir de nove módulos *Inception*. Totalizam cerca de 11 milhões de parâmetros, menos do que a rede AlexNet, apesar de apresentar um desempenho superior.

Modelo utilizado neste trabalho é uma replicação disponível no *model zoo* da Caffe, da GoogleNet original, com diferenças quanto ao treinamento de que não contou com aumentos de base por

algoritmos de reiluminação, mudança de escala ou proporção; com a diferença quanto ao método de inicialização de pesos, utilizando Xavier [48] ao invés do original Gaussiano, além de uma política de decaimento de taxa de aprendizagem diferente. Fora disponibilizada na iteração 2.400.000 com acurácia nominal de 68,7% top1 e 88,9% top5 para o conjunto de validação, aplicando recorte central nas imagens utilizadas para a obtenção desta estatística.

4.2.4 VGG

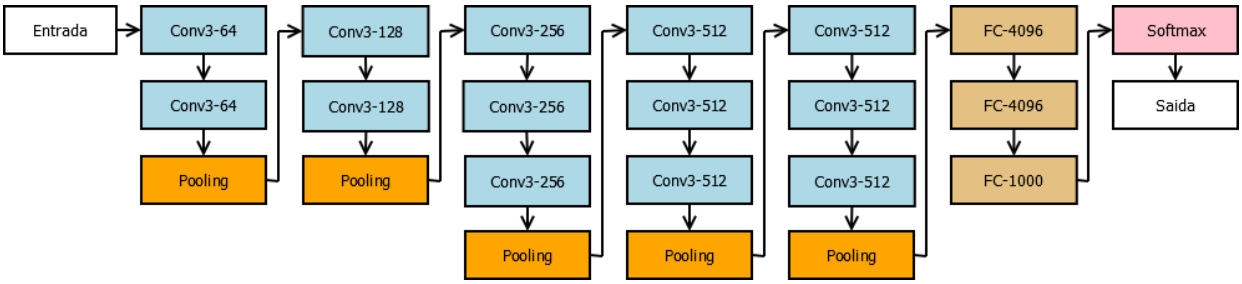


Figura 4.4: Esquema da rede VGG de 16 camadas (com pesos).

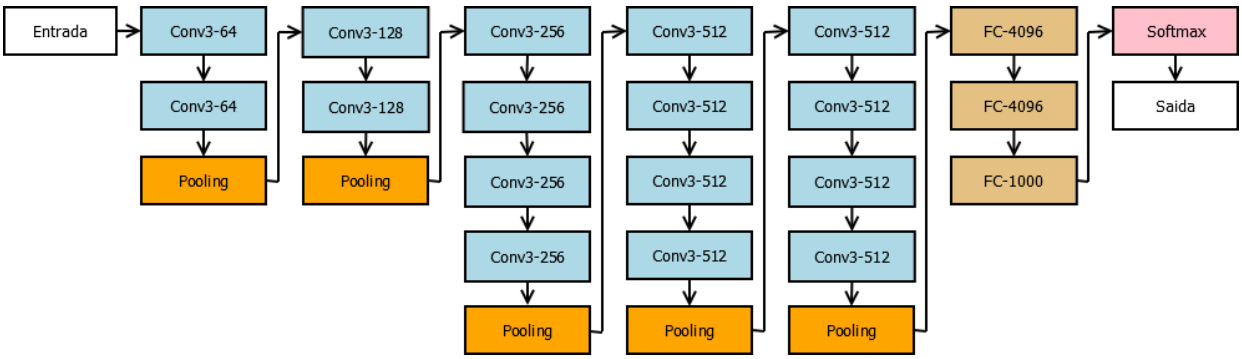


Figura 4.5: Esquema da rede VGG de 19 camadas (com pesos).

Arquiteturas entre as ganhadoras do desafio ILSVRC 2014, criadas pelo grupo VGG (Visual Geometry Group, da Universidade de Oxford), é explicada em detalhes pelo artigo de Simonyan *et al.* [47]. Neste, os autores declaram que as redes, que são ilustradas nas figuras 4.4 e 4.5 e onde unidades ReLU foram omitidas por brevidade, foram projetadas como parte de uma avaliação do impacto da adição de camadas convolucionais em arquiteturas diversas.

Tal avaliação consistiu em estudar a capacidade de diferentes redes em operar como classificadores, construídas de forma sucessiva, em derivações em que se adicionavam novas camadas convolucionais, todas com a restrição espacial de ter dimensão de no máximo 3×3. Em geral, as camadas convolucionais se intercalam com camadas de *pooling* com tamanho de janela 2×2 *pixels* aplicados a passo 2; e são seguidas por três camadas completamente conectadas, em que as duas primeiras possuem 4096 pesos, e a última, 1000. A derradeira camada de todas as redes é uma camada *softmax*.

A motivação por trás da escolha de se utilizar sempre pequenos filtros convolucionais se dá por uma equivalência de campos receptivos eficazes entre camadas convolucionais empilhadas e filtros maiores. Pequenos filtros empilhados resultam na vantagem de que ao se aplicar mais camadas convolucionais, mais não-linearidades são incluídas na arquitetura, o que aumenta o poder discriminatório da mesma. Adicionalmente, a quantidade de pesos é reduzida ao se utilizar pequenos filtros empilhados, em comparação ao uso de filtros simples de maior tamanho. Novamente, de maneira similar e concorrente à GoogleNet, também utilizam convoluções 1×1 em algumas das arquiteturas avaliadas.

A rede de 16 camadas se trata da variante "D"estudada pelos autores, com somente filtros convolucionais de dimensão 3×3, para uma entrada de imagem de dimensões 224×224, é composta do encadeamento de duas camadas em série convolucionais de 64 filtros, uma camada de *pooling*, 2 camadas convolucionais em série de 128 filtros cada, uma camada de *pooling*, 3 camadas convolucionais em série de 256 filtros, uma camada de *pooling*, 3 camadas convolucionais em série de 512 filtros, uma camada de *pooling*, 3 camadas convolucionais em série de 512 filtros, uma última camada de *pooling* e então as camadas completamente conectadas mencionadas anteriormente, da arquitetura básica.

A rede de 19 camadas (variante "E") segue uma estrutura similar à de 16, a que se adiciona uma outra camada convolucional de 256 filtros no antepenúltimo encadeamento de camadas convolucionais, e uma camada de 512 filtros para cada um dos último e penúltimo encadeamentos de camadas convolucionais.

Foram disponibilizados publicamente os 2 melhores modelos testados pré-treinados, dentre 6 principais configurações para a tarefa de classificação. Um contém 16 camadas com pesos e outro 19, sendo que ambas as redes se tratam de versões aprimoradas em relação às utilizadas no ILSVRC 2012. Podem ser acessadas no *model zoo* da Caffe, que inclusive fora utilizado no trabalho dos autores, em uma versão modificada do *framework*.

4.3 Execução

Utilizando das escolhas realizadas durante a fase de levantamentos, segue-se com a aplicação das degradações escolhidas, em diversas variações, em cada imagem da base de dados de validação original do desafio ILSVRC 2012, gerando bases de dados derivadas, únicas para pares de tipo de degradação e variação aplicadas. Deve-se mencionar que esta etapa não se associa a uma tecnologia em específico, podendo ser realizada por programa em C++, Python, Matlab, etc.

Segue com a classificação individual de cada imagem, utilizando o *framework* Caffe. Deve-se ressaltar que enquanto fora feita uma delimitação de tecnologia empregada na elaboração da metodologia, esta escolha permite diversas variações de execução, como o modo que se utiliza o *framework* por programas do mesmo, feitos em C++ ou por script Python, em CPU ou GPU, etc., e fora realizada interdependentemente com o estudo de arquiteturas e técnicas contemporâneas de redes neurais convolucionais.

4.4 Métricas de desempenho

Como métrica a ser utilizada na avaliação do desempenho das redes em sua tarefa de classificação, optou-se por medir a acurácia das classificações, em contraste ao critério de taxa de erro utilizado pelo ILSVRC. Porém, julgou-se interessante a geração de métricas baseadas nos dois tipos de acerto considerados pelo desafio, os chamados "Top1", cuja acurácia correspondente é dada pela equação 4.1

$$A = \frac{\sum_{i=0}^{n} x_i}{n} \quad |x_i = \begin{cases} 1, & \text{para } l_i = r_i \\ 0, & \text{caso contrário} \end{cases} \tag{4.1}$$

em que um acerto é contabilizado quando a maior saída da rede (l_i) à uma entrada é validada como correta, ao compará-la ao rótulo verdadeiro r_i; e "Top5", cuja acurácia correspondente é dada pela equação 4.2

$$A = \frac{\sum_{i=0}^{n} x_i}{n} \quad |x_i = \begin{cases} 1, & \text{para } L_i \supset r_i \\ 0, & \text{caso contrário} \end{cases} \tag{4.2}$$

em que é considerado um acerto quando a classe correta r_i está entre as maiores cinco saídas (L_i) correspondentes à uma entrada. Tal critério é útil pois as imagens utilizadas podem não conter apenas o objeto de sua categoria, de modo que a saída da rede pode considerar mais do que apenas um palpite sobre o objeto dado pelo rótulo da imagem, possivelmente classificando outros objetos nesta mesma, em prioridades diferentes das consideradas pelo classificador manual da base original.

Em ambos os casos, a acurácia é obtida ao se dividir a soma da quantidade de acertos pela quantidade total de amostras n.

Em resumo, a depender do critério de acerto, a métrica principal deste trabalho consiste da razão entre a quantidade de acertos e a quantidade total de imagens testadas, avaliada com a comparação dos rótulos dados pela rede com a tabela verdade da base de dados.

Capítulo 5

Resultados

Este capítulo apresenta os resultados obtidos durante as etapas de execução da metodologia proposta neste trabalho, seguindo uma ordem sequencial, expondo seus resultados por meio de métricas, tabelas e gráficos. Em geral estes são divididos por arquitetura de rede neural testada, ou por categoria de degradação de imagem. Também, expõem aspectos específicos das execuções dos processos propostos pela metodologia.

5.1 Materiais utilizados

Dentre as diversas maneiras possíveis para a geração dos bancos de imagens degradadas, esta fora feita por meio de script construído para o ambiente MATLAB pelo laboratório LISA (Laboratório de Imagens, Sinais e Acústica) da Universidade de Brasília para este trabalho.

A avaliação das bases de imagens nas diferentes redes escolhidas fora feita utilizando o *framework* Caffe, associada à CUDA 7.5.18, para os *drivers* de versão 352.39 em uma NVIDIA GeForce 740m de 2GB, de um notebook Dell Vostro 5470 possuindo também processador Intel(R) Core(TM) i7 @1,80GHz, 8GB de RAM, Linux Mint 17.3 x64 Mate. Apesar da preocupação com performance ser factual, por se tratar de um computador móvel, também havia certa preocupação com a geração de calor de modo que apesar da perda de poder de processamento, optou-se por executar a avaliação para cada imagem, individualmente, utilizando *pycaffe* ao invés de opções de melhor desempenho.

5.2 Geração dos bancos de imagens

Todos os processos de geração de bases de imagens degradadas, por problemas com cabeçalho das imagens de referência, resultaram em erros de processamento. Decorrente destes erros, todas as bases de imagens degradadas, exceto as de compressões JPEG 2000 contém apenas 49.100 imagens. As bases JPEG 2000 em específico contém 49.038 imagens cada.

Por este motivo, ao invés de considerar para efeitos de comparação e ordenamento, um cálculo

único das medidas de acurácia para cada combinação de base de imagens e de redes convolucionais, calculou-se também as medidas para um subconjunto da base de imagens original correspondente ao conjunto de imagens degradadas testado, de modo que os resultados passam a ser dados em pares: acurácia da base degradada e acurácia do subconjunto equivalente da base original, conforme descritas na Seção 4.4.

5.3 Resultados das classificações

A análise da acurácia das redes selecionadas consistiu em avaliar cada imagem de cada tipo de degradação, em cada rede, salvando as cinco maiores respostas em arquivo de texto. Com os dados obtidos, um *script* em linguagem *Python* realiza a comparação para cada resultado com os valores de uma tabela verdade, gerando valores de acurácia por classe de degradação por rede utilizada.

Tais resultados podem ser observados nas Tabelas II.1, II.2, II.3, II.4, II.5 do Anexo II deste documento. Alternativamente, os dados de todas as degradações agrupados por arquitetura de rede convolucional, ordenados por acurácia podem ser visualizados nas Figuras 5.1 à 5.10, em que a linha contínua representa as acurácias dos subconjuntos da base original, enquanto a linha tracejada, os dados das bases degradadas. No Anexo I se encontram gráficos gerados separados por categoria e ordenados por rótulo.

De modo geral, a influência das degradações testadas deverá ser avaliada pela comparação entre as métricas empregadas, as acurácias Top1 e Top5, e valores de referência obtidos pelo cálculo destas mesmas métricas utilizando a base de imagens original, sem degradação, como entrada. Tais valores de referência calculados com imagens da base original podem ser visualizados na Tabela 5.1.

	AlexNet	CaffeNet	GoogleNet	VGG16	VGG19
Top1	55.80	56.03	68.02	65.77	66.15
Top5	79.13	79.39	88.48	86.65	86.95

Tabela 5.1: Valores de acurácia obtidos para a base de imagens original, em %.

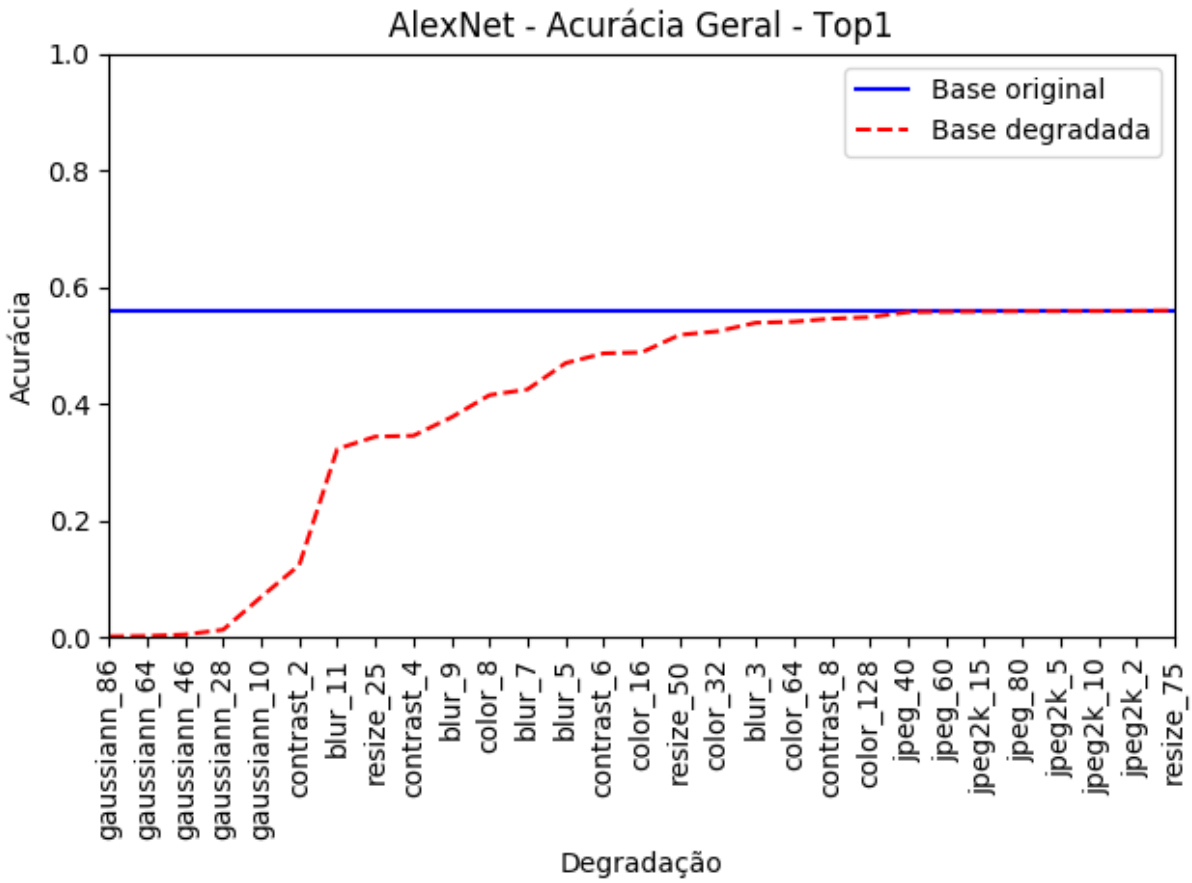


Figura 5.1: Acurácia geral calculada para a rede AlexNet - (Top1).

Em que os rótulos são dados no formato "Tipo de Degradação"_"Variante", que se traduz como "**Tipo**: Descrição. Detalhamento das variantes": **"gaussiann"**: Ruído Gaussiano. A numeração da variante representa o desvio padrão do ruído gaussiano aplicado, em centésimos. **"blur"**: Desfoque (blur). A numeração da variante representa o tamanho do filtro gaussiano utilizado. **"contrast"**: Redução do Contraste. A numeração da variante representa a porcentagem da parte inferior da faixa de intensidades de cor que foi mapeada para a faixa completa, em centésimos. **"resize"**: Redução do tamanho espacial. A numeração da variante representa para qual porcentagem a imagem foi redimensionada. **"color"**: Redução dos níveis de cores disponíveis para formar a imagem, por canal. A numeração da variante representa quantas. **"jpeg"**: Compressão JPEG. A numeração da variante é parâmetro "Qualidade"(Quality) utilizado pelo compressor. **"jpeg2k"**: Compressão JPEG 2000. A numeração da variante é parâmetro "Taxa de Compressão"(CompressionRatio) do compressor.

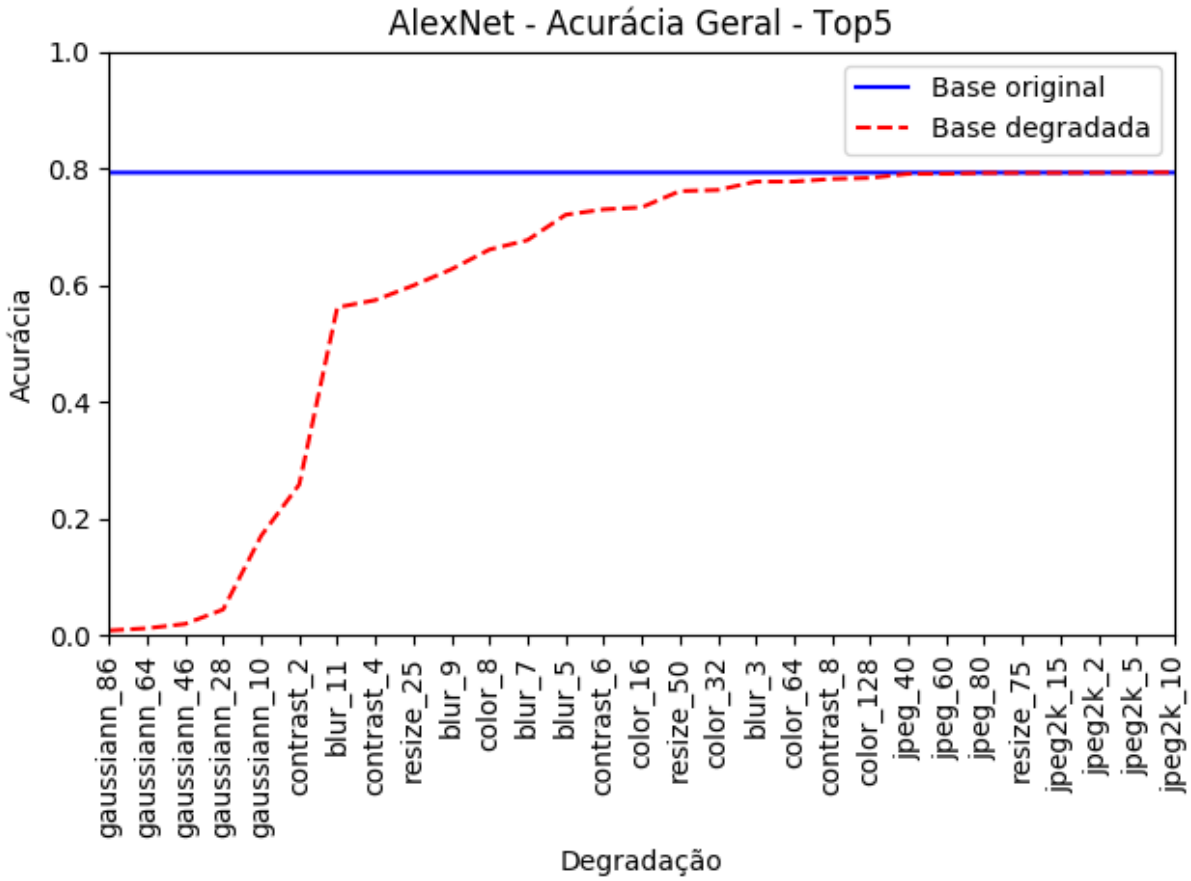


Figura 5.2: Acurácia geral calculada para a rede AlexNet (Top5).

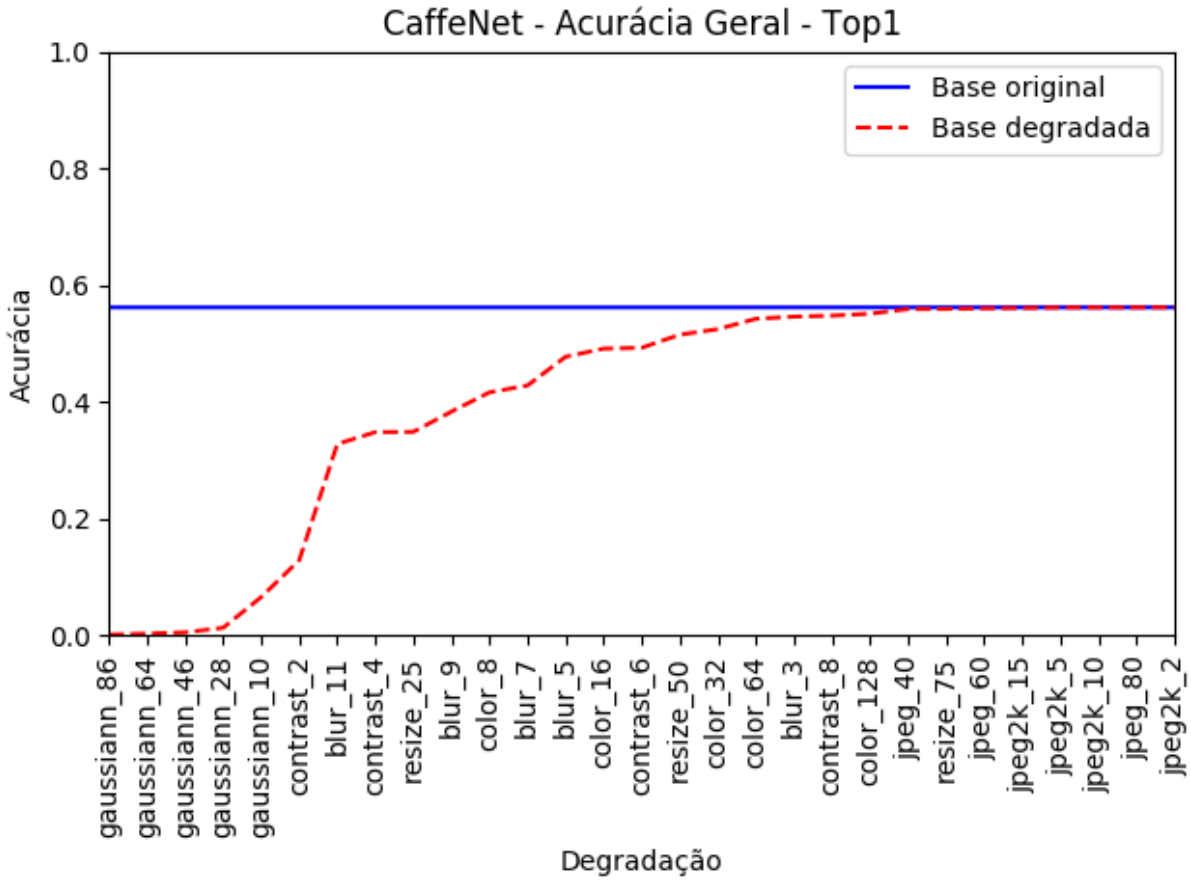


Figura 5.3: Acurácia geral calculada para a rede Caffe (Top1).

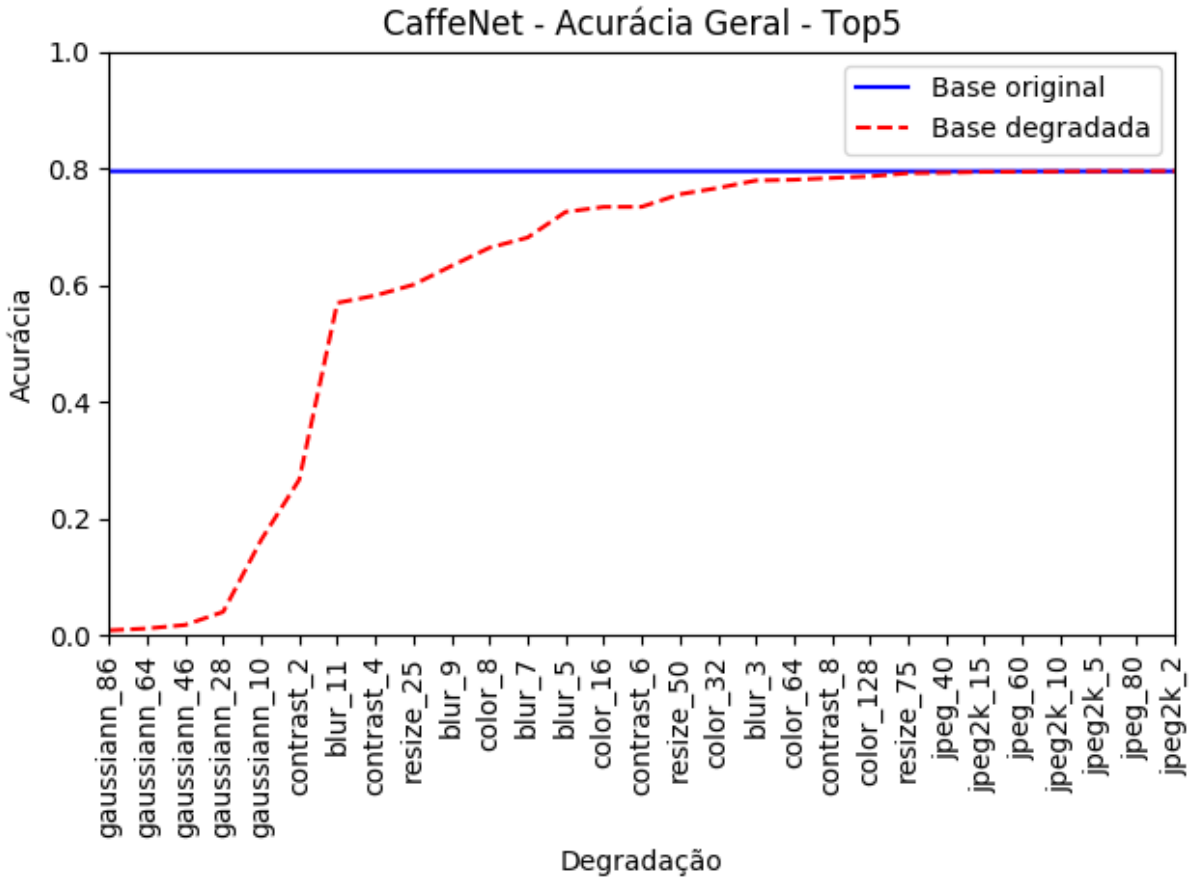


Figura 5.4: Acurácia geral calculada para a rede Caffe (Top5).

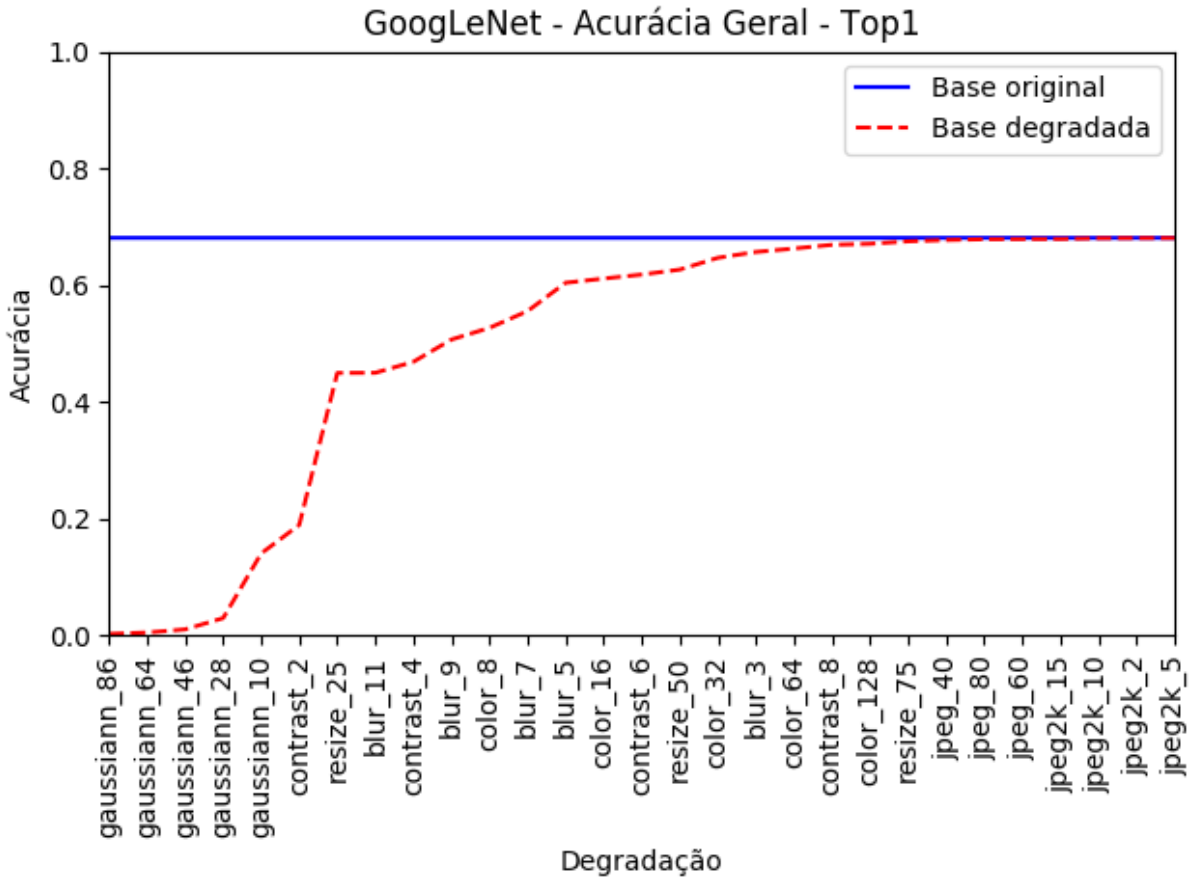


Figura 5.5: Acurácia geral calculada para a rede GoogLeNet (Top1).

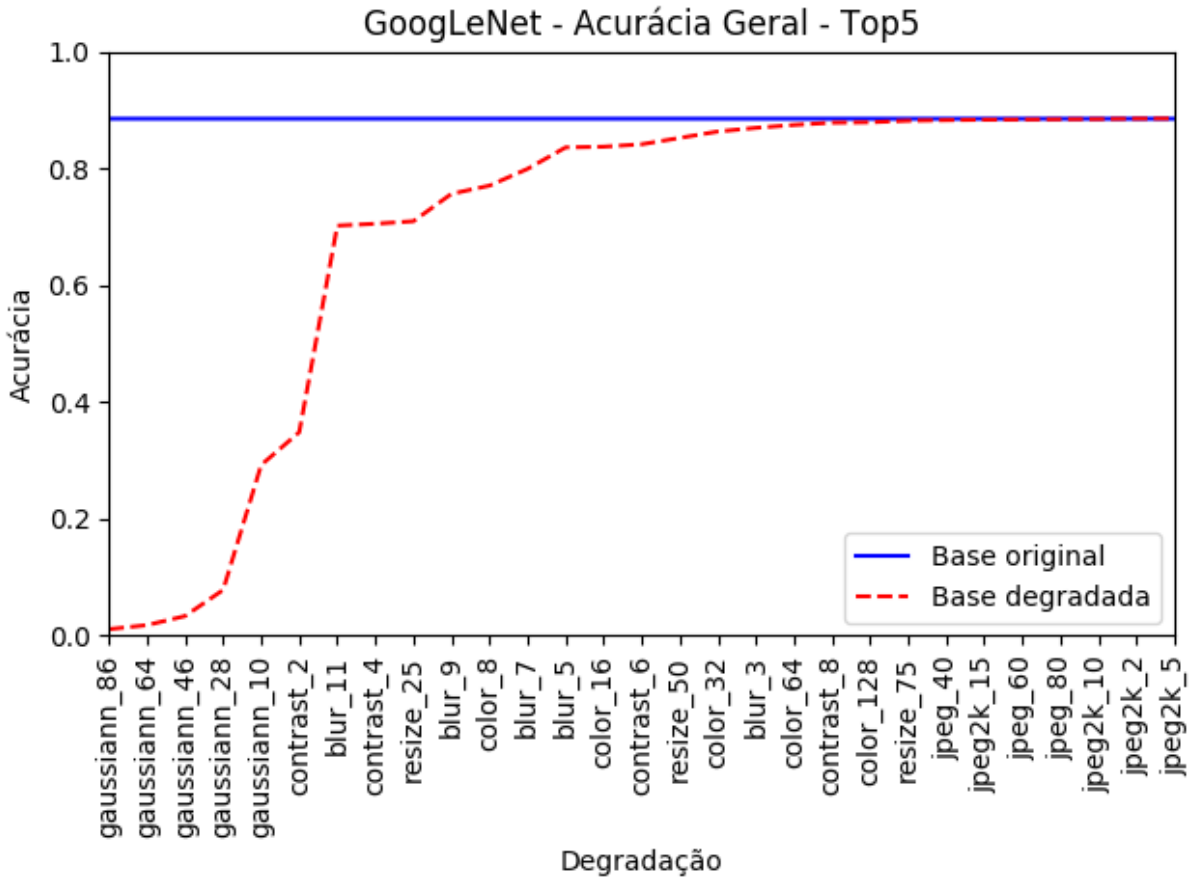


Figura 5.6: Acurácia geral calculada para a rede GoogLeNet (Top5).

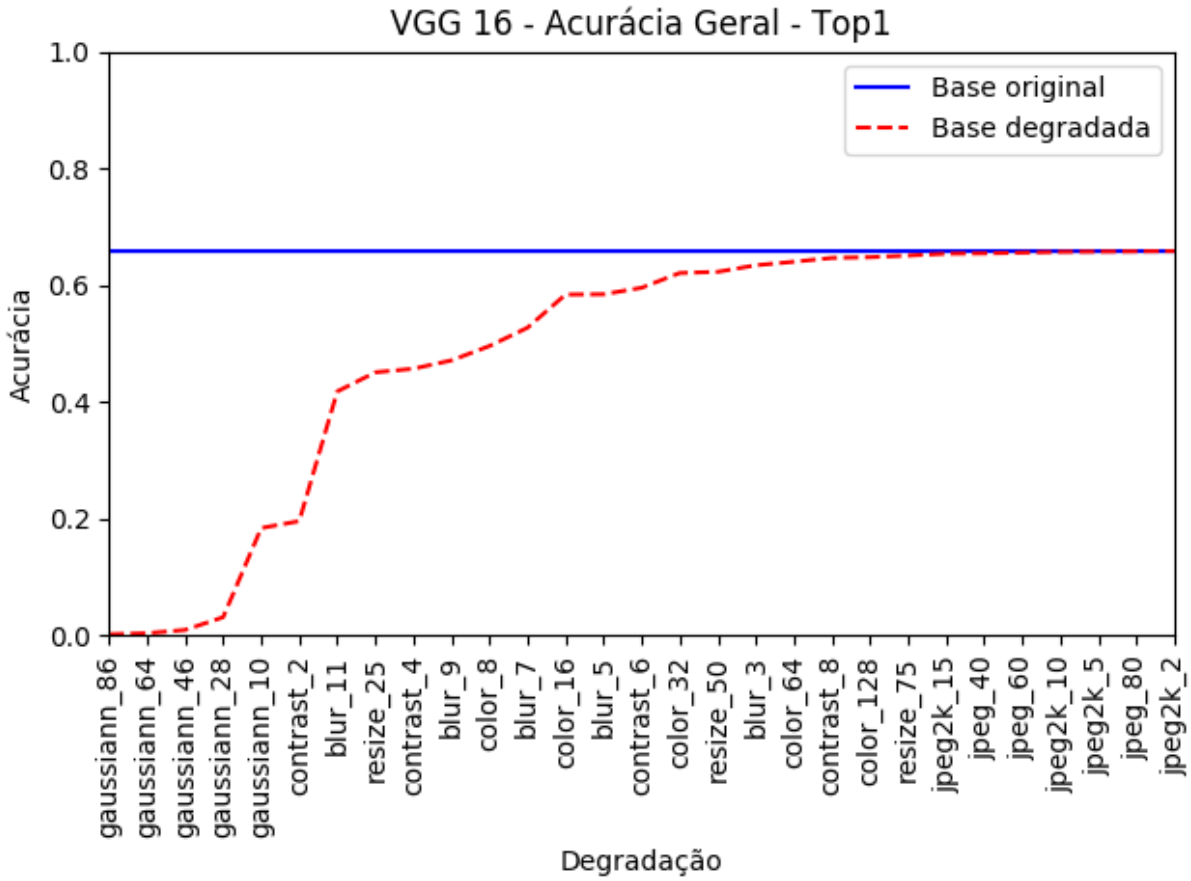


Figura 5.7: Acurácia geral calculada para a rede VGG 16 (Top1).

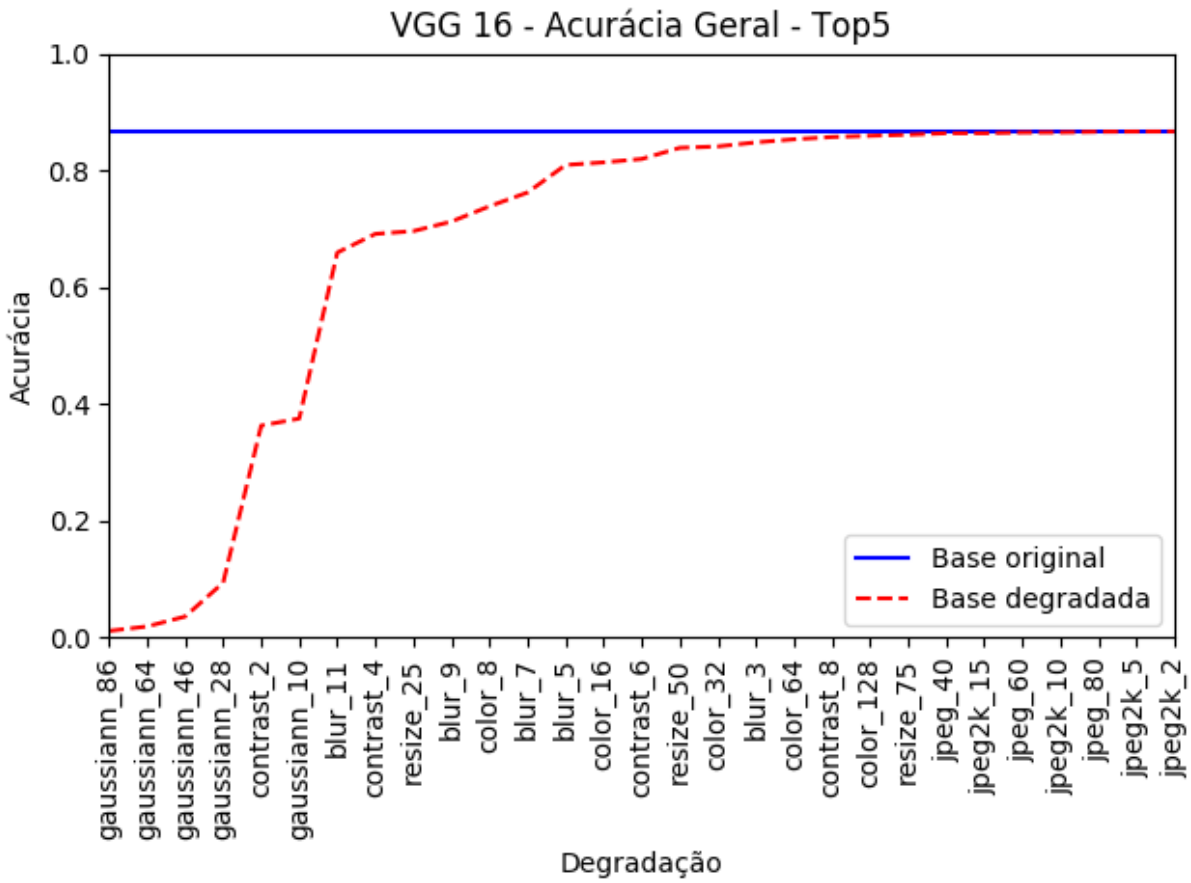


Figura 5.8: Acurácia geral calculada para a rede VGG 16 (Top5).

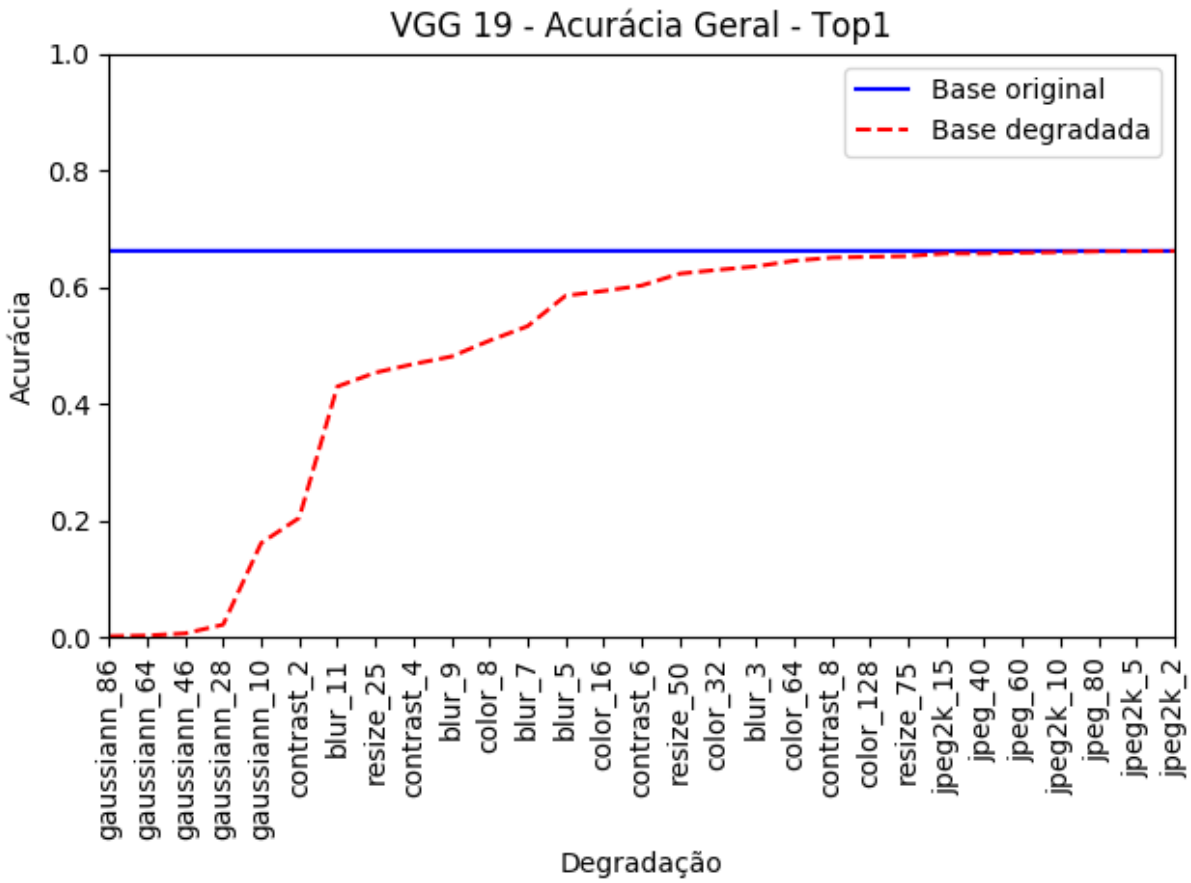


Figura 5.9: Acurácia geral calculada para a rede VGG 19 (Top1).

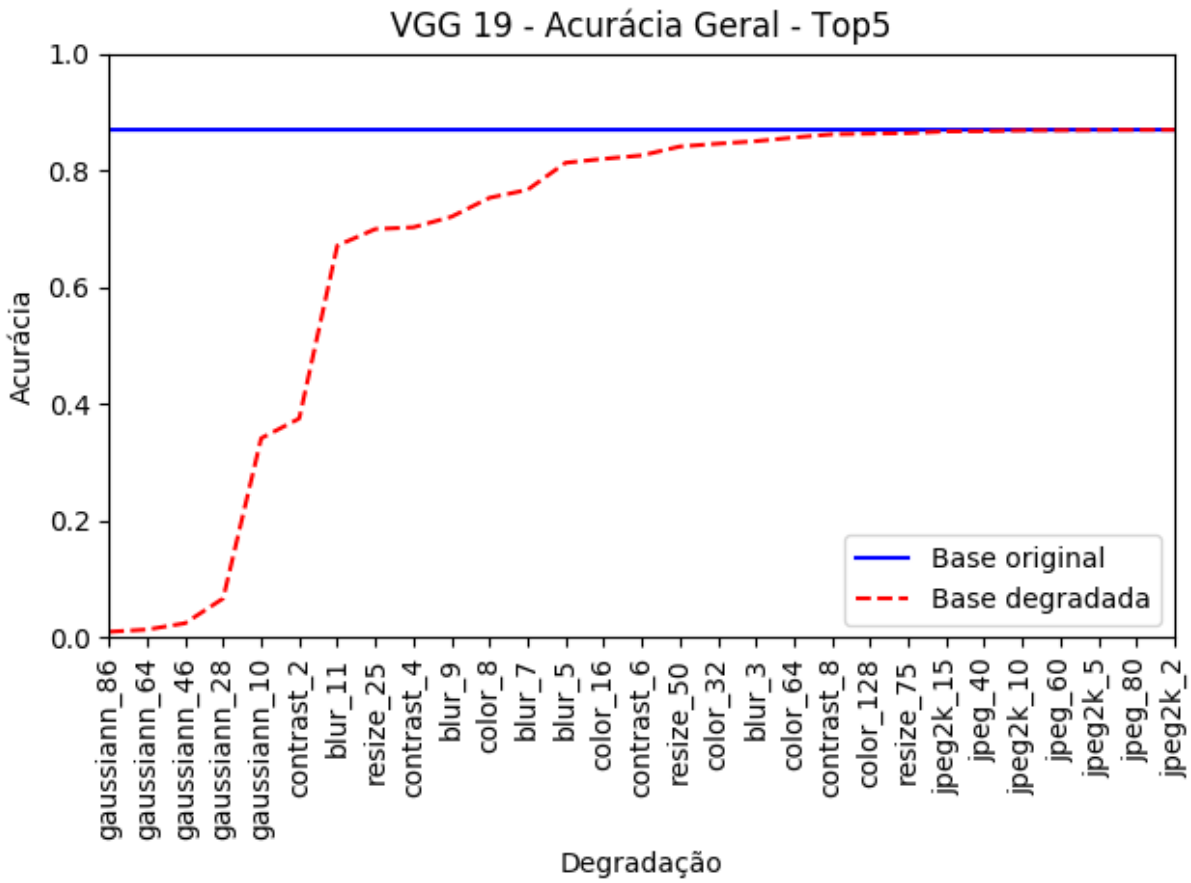


Figura 5.10: Acurácia geral calculada para a rede VGG 19 (Top5).

Primeiramente, observa-se que houve pouca alteração, em geral inferior a 1%, nos valores de acurácia obtidos para os subconjuntos da base de validação original correspondentes aos conjuntos de imagens degradadas, quando comparados aos valores de referências demonstrados na tabela 5.1. Estes valores são representados pela linha azul contínua no gráfico e estão disponíveis no anexo II.

Esta pouca alteração fora utilizada como justificativa para não se empregar uma proporção entre a acurácia da base degradada e a acurácia da base equivalente como critério de ordenamento do eixo das abcissas nas figuras 5.1 à 5.10, que é feito apenas pela ordem crescente de acurácias para as bases degradadas. A pequena magnitude da variação atesta a pequena relevância das imagens que apresentaram problemas no processo de degradação na construção destes parâmetros de acurácia.

Em se tratando da avaliação dos resultados obtidos entre mesmas categorias de degradações estudadas, indica-se as imagens do Anexo I, no qual as Figuras I.1 e I.2 apresentam respectivamente as acurácias Top1 e Top5 calculadas para as diferentes redes empregadas com imagens degradadas por desfoque (blur), e de maneira similar, as Figuras I.3 e I.4 representam a redução de cores, as Figuras I.5 e I.6 representam a redução de contraste, as Figuras I.7 e I.8 a aplicação de ruído, as Figuras I.9 e I.10 a compressão JPEG, as Figuras I.11 e I.12 a compressão JPEG2000 e por fim, as Figuras I.13 e I.14 apresentam os resultados correspondentes à redução de escala das imagens,

todas expressando acurácias Top1 e Top5 respectivamente.

Pode-se notar que as redes foram especialmente sensíveis ao ruído e ao desfoque (*blur*). Apesar de possuir certas propriedades de desfoque, nota-se que as redes se mostraram especialmente resistentes aos processos de compressão JPEG e JPEG 2000. Vale observar que enquanto as bases de imagens ruidosas, como na figura 3.4, se mostram em geral subjetivamente degradadas de modo severo, pouco se observa degradações nas imagens das bases de compressões, como nas Figuras 3.5 e 3.6. Entretanto, tal resultado concorda com os obtidos por Dodge e Karam [31]. O mesmo não pode ser dito sobre os resultados obtidos para a degradação de contraste, que enquanto neste trabalho fora feito pelo mapeamento de intensidades, na referência fora feita pela mesclagem das imagens com uma imagem cinzenta. Em nossos resultados, a variação do contraste apresentou variação da resposta equivalente à variação para o desfoque (*blur*) da imagem, estando bastante acentuada.

Todas redes também se mostraram bastante sensíveis ao redimensionamento e às degradações de cor, e em especial ao primeiro tipo de degradação. Apesar da crescente resolução das imagens geradas nas mais diversas aplicações, ainda há aquelas em que baixa resolução se mostra comum e as redes neurais convolucionais aqui utilizadas podem ser adaptadas para utilização em outras aplicações com o uso de mecanismos de *Transfer Learning*. Condições de espaço de cor reduzido também se mostram importantes quando na utilização de hardware específico, especialmente antigo, apesar de serem menos comuns que variações de dimensionalidade e apresentarem menor sensibilidade.

Em se tratando de causas para a sensibilidade destas degradações, deve-se mencionar que uma vez que o tamanho de imagens de entrada das redes convolucionais utilizadas é fixo, imagens pequenas deverão ser redimensionadas (e recortadas) para tamanho compatível com a entrada, utilizando processos de interpolação. A imagem apresentada à rede se mostrará como desfocada, sem grandes detalhes de textura ou de bordas, possivelmente necessários para boa operação das CNNs. Quanto à redução da disponibilidade de cores, esta discretiza pequenas variações de texturas ou pequenas transições na imagem em tonalidades idênticas, eliminando a diferenciações possivelmente relevantes para as redes avaliadas.

Quanto às variações por arquitetura, de imediato percebe-se que o aumento de profundidade resulta em uma diminuição da sensibilidade às degradações. Nos gráficos (Figuras 5.1 a 5.10) em geral é possível perceber não uma mera escala da curva de acurácias das bases de imagens degradadas, mas que estas são mais acentuadas para as arquiteturas mais profundas. Possivelmente isto se deve à geração de abstrações mais complexas e menos sujeitas a degradações nas camadas hierarquicamente superiores das redes profundas em contraste às das rasas.

Capítulo 6

Conclusões

Este trabalho, em seus objetivos de geração de bases de imagens degradadas pelo uso e o estudo de redes neurais convolucionais de destaque, teve sucesso em sua realização. Apesar de problemas de processamento relacionados à aplicação de distorções de qualidade nas imagens de referência, ainda foram obtidas bases consideravelmente maiores que as utilizadas em trabalhos relacionados, como o de Dodge [31]. Não houveram problemas em obter e executar CNNs do estado-da-arte para a obtenção de dados a serem processados e avaliados.

O procedimento proposto para a avaliação do desempenho de redes neurais convolucionais classificatórias quando expostos a imagens degradadas como entrada também fora bem-sucedido, com resultados que demonstraram a queda de performance com o aumento de degradação condizentes com o esperado e na medida em que se faz uma comparação adequada, com trabalhos relacionados como os de Dodge [31] e Karam [22].

Com este trabalho se demonstra que aplicações com severas restrições de hardware, que implicassem na redução do espaço de cores, de dimensionalidade das imagens, em imagens ruidosas ou mesmo desfocadas teriam uma operação de softwares que utilizassem CNNs severamente dificultada, ao menos para arquiteturas como as testadas, construídas sem a consideração destas limitações.

Também sugere que arquiteturas mais profundas de redes neurais convolucionais apresentam sensibilidade ligeiramente menor a degradações de imagem, o que se mostra outro aspecto importante a ser considerado na aplicação de CNNs como método, ligado a questões de capacidade de memória e de capacidade de processamento empregados. Não necessariamente o aumento de profundidade de CNNs representará uma maior exigência destes requisitos, como atesta a rede GoogleNet, em comparação à AlexNet, que é mais profunda, mas utiliza menos parâmetros, e, portanto, menos memória; mas o bom desempenho de redes mais profundas deve ser tomado juntamente aos outros critérios citados, como fator de seleção de CNNs para seu uso como ferramenta.

Mas para além de qualquer sugestão de restrição ou aplicação que se possa citar neste documento, as degradações de imagem estudadas se fazem presentes em situações comuns e fatuais, de modo que se espera que este documento se prove eficaz quando na seleção de redes convolucionais como fator presente em um sistema que opere com estas degradações como característica, ao

demonstrar como estas degradações afetam o desempenho de arquiteturas de referência.

6.1 Perspectivas Futuras

É possível notar que os parâmetros utilizados para a determinação do nível de degradação de cada base de imagens dentro de um mesmo conjunto de degradações por vezes resultou em variações de acurácia que se concentraram próximos ao zero, como para nas bases de ruído gaussiano, ou à valores que as redes apresentariam em presença de imagens sem degradação, como nas bases de compressões, ao que de imediato pode ser proposto que o conjunto de bases de imagens fosse expandido para obter uma distribuição mais uniforme. Neste caso, sugere-se a criação de uma base de dados com degradações combinadas e avaliar em novos testes esta influência.

Também, apenas se demonstra o efeito das degradações de diminuir o desempenho das CNNs, somente sugerindo, mas sem explicar especificamente quais os fatores responsáveis por este efeito. Nesta situação poderia ser realizado um estudo dos efeitos das degradações de imagens na estrutura interna das redes neurais convolucionais, avaliando o tipo de características reconhecidas pelos filtros, como são afetadas pelas degradações, e como estes efeitos se propagam pela hierarquia da rede.

Outras métricas além da acurácia poderiam ser estudadas ou até mesmo elaboradas, de modo a realizar um estudo da relação entre a qualidade percebida pelas redes neurais e a qualidade efetiva, possivelmente permitindo inclusive que fossem realizadas comparações entre degradações aplicadas, ajudando a explicar comportamentos e características das redes quanto a similaridades ou diferenças entre as degradações.

Por último, um próximo passo seria sair de um escopo de somente avaliação para um de treinamento de redes neurais convolucionais profundas, de modo a estudar a possibilidade de geração de redes mais robustas a degradações de imagem por meio de aumentos nas bases de dados de treinamento, ou pela alteração de aspectos de arquitetura das redes, escolhendo estruturas menos suscetíveis às degradações.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] TRENDS, G. *convolutional neural networks - Explorar - Google Trends*. Acessado em: 29/06/2017. Disponível em: <https://trends.google.com/trends/explore?date=all&q=convolutional%20neural%20networks>.

[2] VIOLA, P.; JONES, M. J. Robust real-time face detection. *International Journal of Computer Vision*, v. 57, n. 2, p. 137–154, May 2004. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb>.

[3] MORDVINTSEV, A.; OLAH, C.; TYKA, M. *Inceptionism: Going Deeper into Neural Networks*. Disponível em: <https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.

[4] GATYS, L. A.; ECKER, A. S.; BETHGE, M. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. Disponível em: <http://arxiv.org/abs/1508.06576>.

[5] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., p. 1097–1105, 2012. Disponível em: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.

[6] DENG, J. et al. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009.

[7] MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <http://dx.doi.org/10.1007/BF02478259>.

[8] ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.

[9] WERBOS, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tese (Doutorado) — Harvard University, 1974.

[10] FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, v. 36, p. 193–202, 1980.

[11] HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, v. 195, p. 215–243, 1968. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1557912/pdf/jphysiol01104-0228.pdf>.

[12] LECUN, Y.; BENGIO, Y. The handbook of brain theory and neural networks. In: ARBIB, M. A. (Ed.). Cambridge, MA, USA: MIT Press, 1998. cap. Convolutional Networks for Images, Speech, and Time Series, p. 255–258. ISBN 0-262-51102-9. Disponível em: <http://dl.acm.org/citation.cfm?id=303568.303704>.

[13] PATTERSON, D. A.; HENNESSY, J. L. *Computer Organization and Design, Fifth Edition: The Hardware/Software Interface*. 5th. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN 0124077269, 9780124077263.

[14] RUPP, K. *CPU, GPU and MIC Hardware Characteristics over Time*. Acessado em: 01/07/2017. Disponível em: <https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/>.

[15] PARIKH, J. *Velocity 2012: Jay Parikh, "Building for a Billion Users"*. Video. Acessado em: 01/07/2017. Disponível em: <https://youtu.be/oodS71YtkGU>.

[16] CHAN, C. *What Facebook Deals with Everyday: 2.7 Billion Likes, 300 Million Photos Uploaded and 500 Terabytes of Data*. Acessado em: 01/07/2017. Disponível em: <http://gizmodo.com/5937143/what-facebook-deals-with-everyday-27-billion-likes-300-million-photos-uploaded-and-500-terabytes-of-data>.

[17] WOJCICKI, S. *Industry Keynote with YouTube CEO Susan Wojcicki (VidCon 2015)*. Video. Disponível em: <https://youtu.be/O6JPxCBlBh8>.

[18] MORENO, H. *The Importance Of Data Quality – Good, Bad Or Ugly*. Acessado em: 01/07/2017. Disponível em: <https://www.forbes.com/sites/forbesinsights/2017/06/05/the-importance-of-data-quality-good-bad-or-ugly>.

[19] ZOU, W. W. W.; YUEN, P. C. Very low resolution face recognition problem. *IEEE Transactions on Image Processing*, v. 21, n. 1, p. 327–340, Jan 2012. ISSN 1057-7149.

[20] REN, C. X.; DAI, D. Q.; YAN, H. Coupled kernel embedding for low-resolution face image recognition. *IEEE Transactions on Image Processing*, v. 21, n. 8, p. 3770–3783, Aug 2012. ISSN 1057-7149.

[21] KRIZHEVSKY, A.; HINTON, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[22] KARAM, L. J.; ZHU, T. Quality labeled faces in the wild (qlfw): a database for studying face recognition in real-world environments. *Proc. SPIE*, v. 9394, p. 93940B–93940B–10, 2015. Disponível em: <http://dx.doi.org/10.1117/12.2080393>.

[23] TAO, J.; HU, W.; WEN, S. Multi-source adaptation joint kernel sparse representation for visual classification. *Neural Netw.*, Elsevier Science Ltd., Oxford, UK, UK, v. 76, n. C, p. 135–151, abr. 2016. ISSN 0893-6080. Disponível em: <http://dx.doi.org/10.1016/j.neunet.2016.01.008>.

[24] BASU, S. et al. Learning sparse feature representations using probabilistic quadtrees and deep belief nets. *CoRR*, abs/1509.03413, 2015. Disponível em: <http://arxiv.org/abs/1509.03413>.

[25] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998. ISSN 0018-9219.

[26] SHEIKH, H. R.; SABIR, M. F.; BOVIK, A. C. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, v. 15, n. 11, p. 3440–3451, Nov 2006. ISSN 1057-7149.

[27] BOUZERDOUM, A.; HAVSTAD, A.; BEGHDADI, A. Image quality assessment using a neural network approach. In: *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.* [S.l.: s.n.], 2004. p. 330–333.

[28] KANG, L. et al. Convolutional neural networks for no-reference image quality assessment. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1733–1740. ISSN 1063-6919.

[29] Cadieu, C. F. et al. Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Computational Biology*, v. 10, p. e1003963, dez. 2014.

[30] ULLMAN, S. et al. Atoms of recognition in human and computer vision. *PNAS*, v. 113, p. 2744–2749, 03/2016 2016. ISSN 1091-6490. Disponível em: <http://www.pnas.org/content/113/10/2744.abstract>.

[31] DODGE, S.; KARAM, L. Understanding how image quality affects deep neural networks. In: *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*. [S.l.: s.n.], 2016. p. 1–6.

[32] GOODFELLOW, I. J.; SHLENS, J.; SZEGEDY, C. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. Disponível em: <http://arxiv.org/abs/1412.6572>.

[33] Karahan, S. et al. How image degradations affect deep cnn-based face recognition? ago. 2016. Disponível em: <https://arxiv.org/abs/1608.05246>.

[34] GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

[35] ITU. *ISO/IEC 10918-1 : 1993(E) CCIT Recommendation T.81*. 1993. Disponível em: <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>.

[36] ITU. *ISO/IEC15444-1: Information technology— JPEG 2000 image coding system: Core coding system*. 2000.

[37] GUYTON, A. C.; HALL, J. E. *Tratado de Fisiologia Médica*. 12. ed. [S.l.]: Elsevier, 2011. ISBN 8535237356.

[38] LEE, C. *Visual comparison of convolution, cross-correlation and autocorrelation of two signals*. Imagem. Acessado em: 01/07/2017. Disponível em: <https://commons.wikimedia.org/wiki/File Comparison_convolution_correlation_de.svg>.

[39] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. `http://www.deeplearningbook.org`.

[40] RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. 2015. Disponível em: <https://arxiv.org/abs/1409.0575>.

[41] ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <http://tensorflow.org/>.

[42] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, maio 2016. Disponível em: <http://arxiv.org/abs/1605.02688>.

[43] COLLOBERT, R.; KAVUKCUOGLU, K.; FARABET, C. Torch7: A matlab-like environment for machine learning. In: *BigLearn, NIPS Workshop*. [S.l.: s.n.], 2011.

[44] JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[45] DONAHUE, J. *BAIR/BVLC CaffeNet Model*. Acessado em: 01/07//2017. Disponível em: <https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet>.

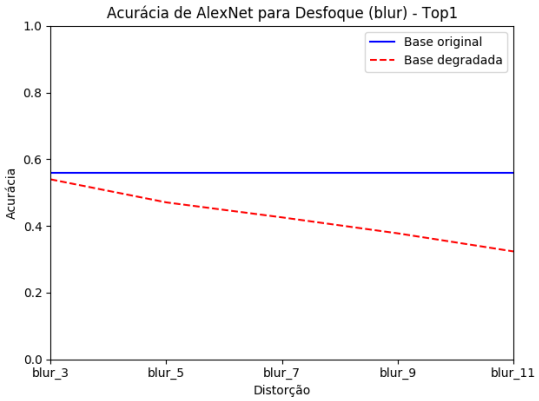
[46] SZEGEDY, C. et al. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. Disponível em: <http://arxiv.org/abs/1409.4842>.

[47] SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <http://arxiv.org/abs/1409.1556>.

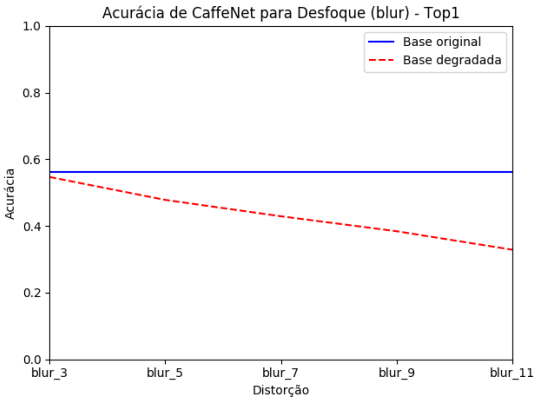
[48] GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*. [S.l.: s.n.], 2010.

ANEXOS

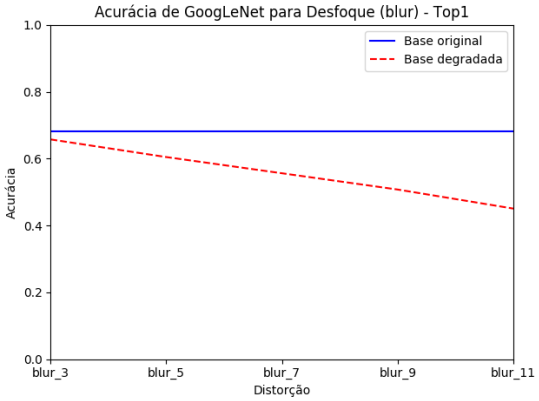
I. GRÁFICOS



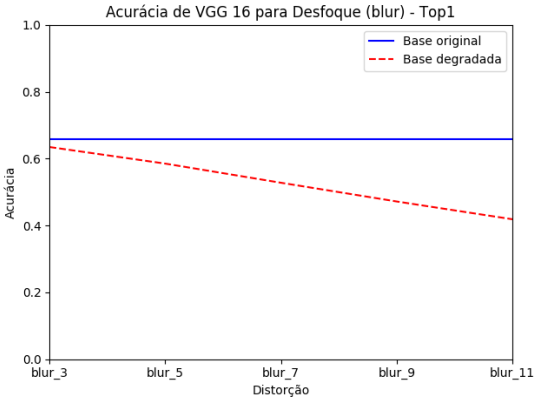
(a) Rede AlexNet



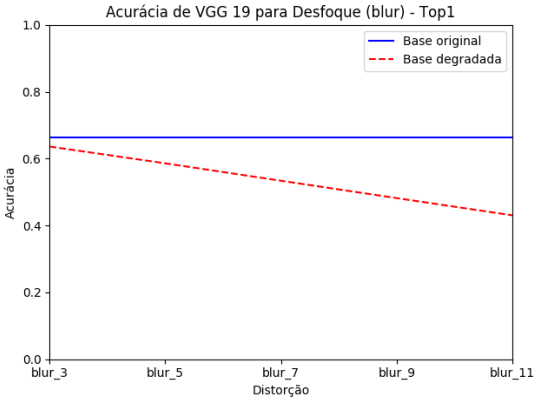
(b) Rede CaffeNet



(c) Rede GoogleNet

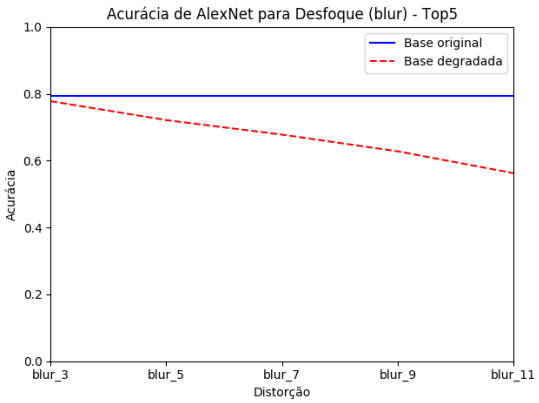


(d) Rede VGG de 16 Camadas

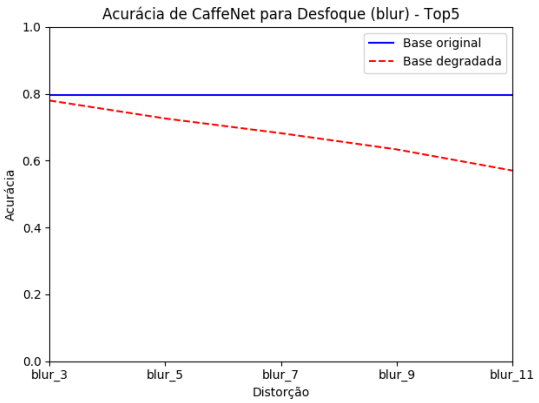


(e) Rede VGG de 19 Camadas

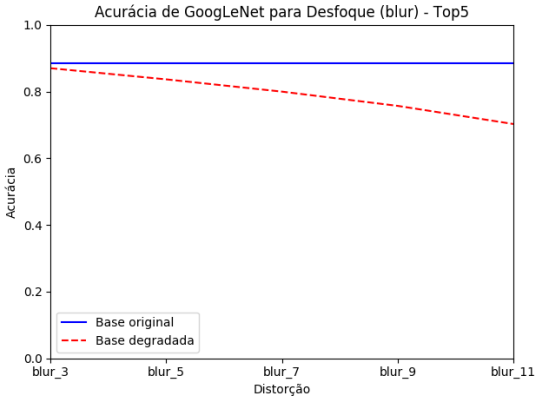
Figura I.1: Acurácia Top1 das redes avaliadas para desfoque (blur).



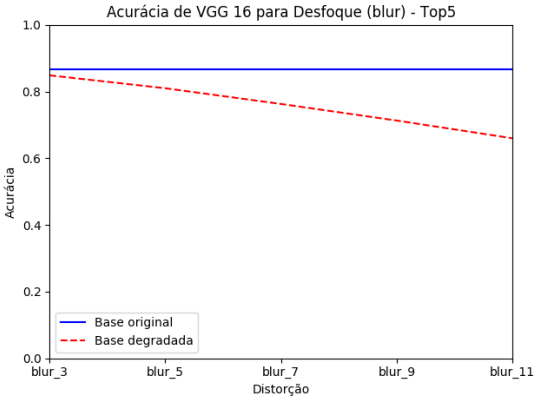
(a) Rede AlexNet



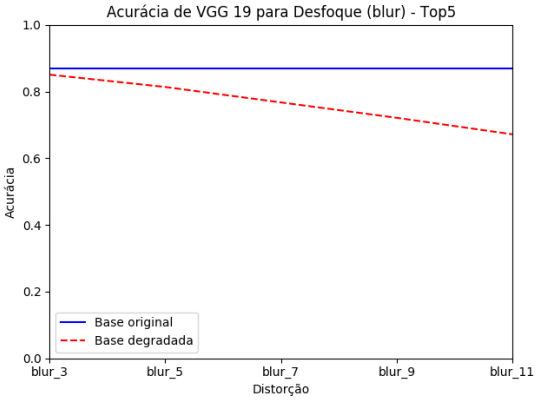
(b) Rede CaffeNet



(c) Rede GoogleNet

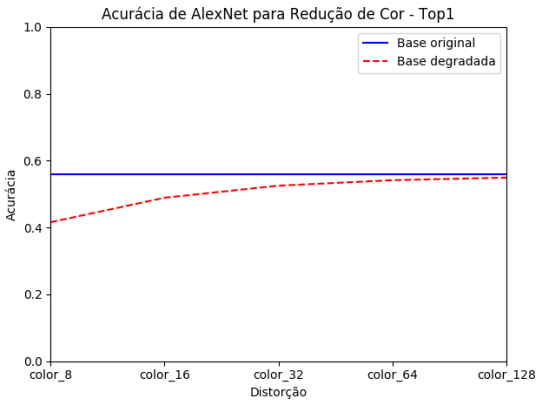


(d) Rede VGG de 16 Camadas

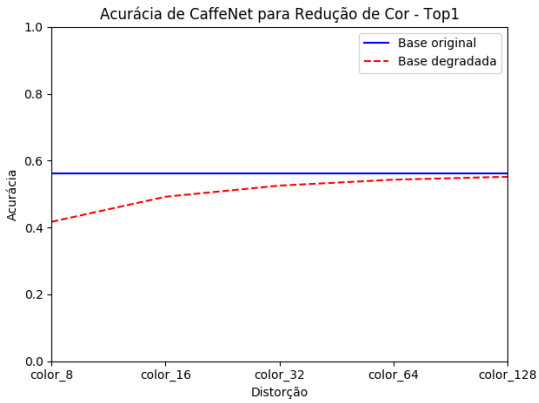


(e) Rede VGG de 19 Camadas

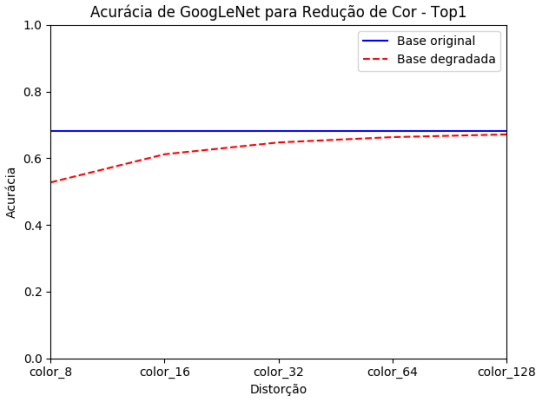
Figura I.2: Acurácia Top5 das redes avaliadas para desfoque (blur).



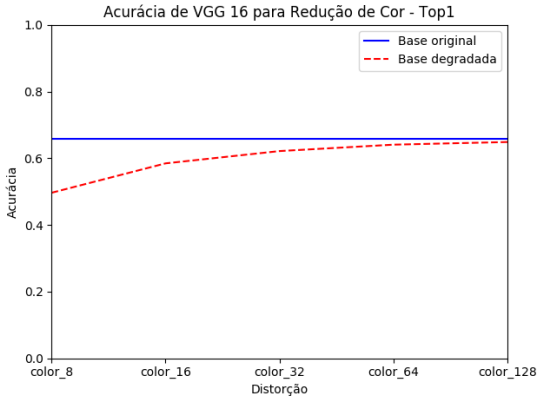
(a) Rede AlexNet



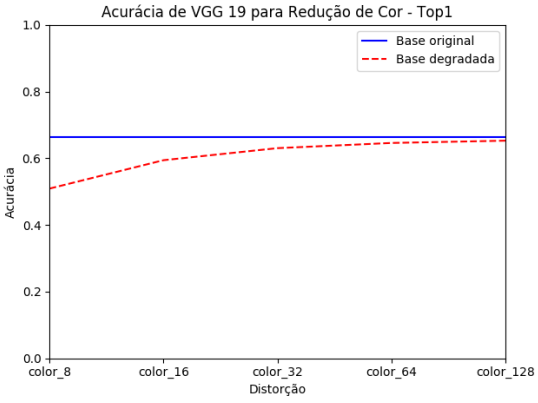
(b) Rede CaffeNet



(c) Rede GoogleNet

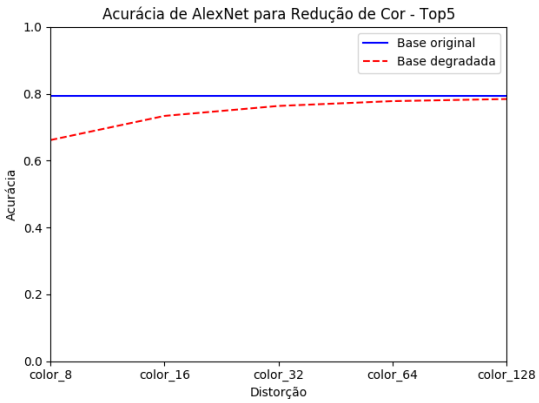


(d) Rede VGG de 16 Camadas

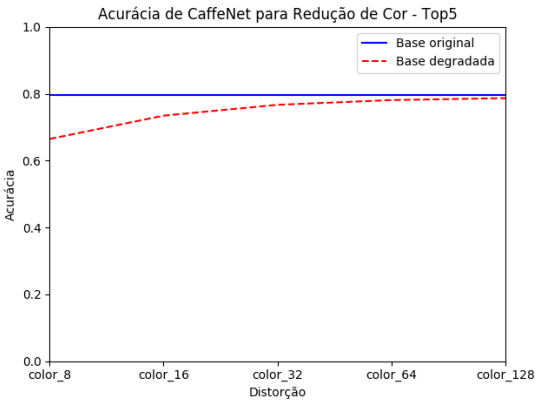


(e) Rede VGG de 19 Camadas

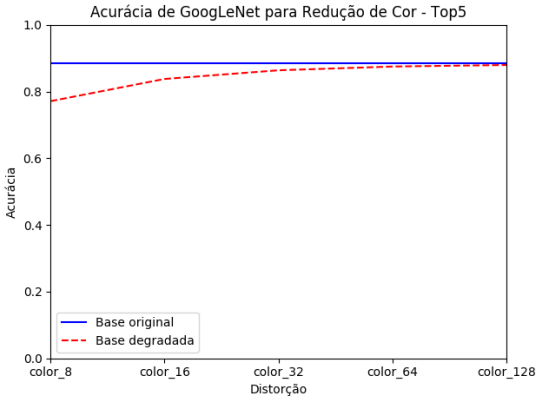
Figura I.3: Acurácia Top1 das redes avaliadas para redução do espaço de cores.



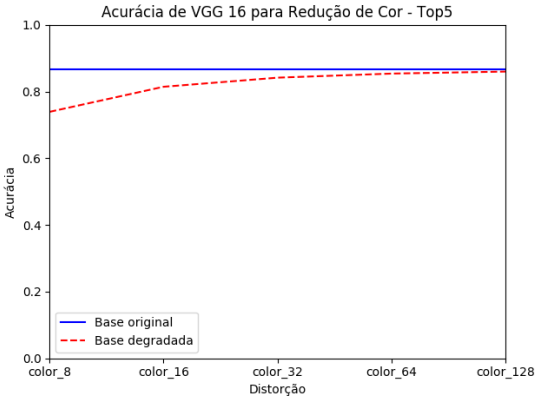
(a) Rede AlexNet



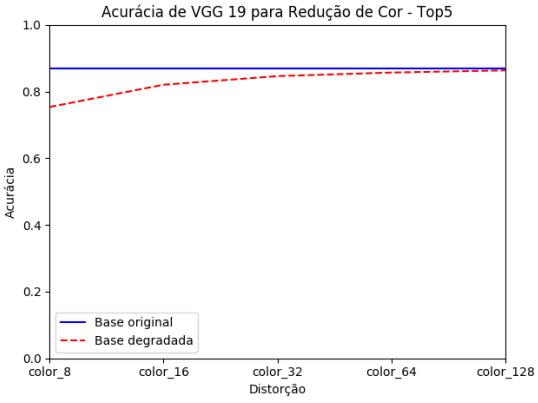
(b) Rede CaffeNet



(c) Rede GoogleNet

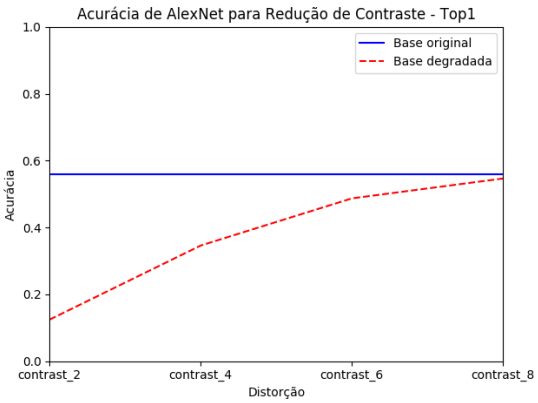


(d) Rede VGG de 16 Camadas

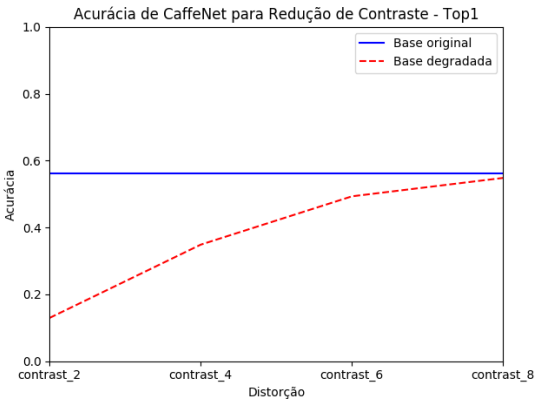


(e) Rede VGG de 19 Camadas

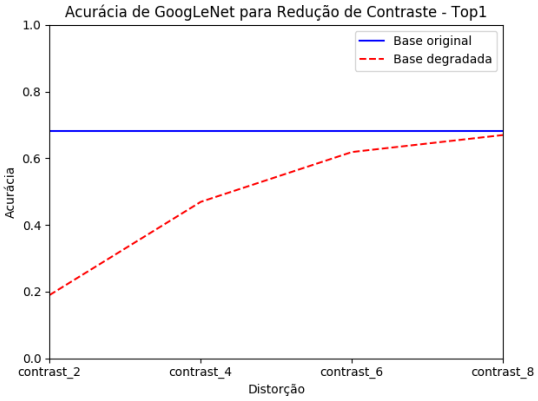
Figura I.4: Acurácia Top5 das redes avaliadas para redução do espaço de cores.



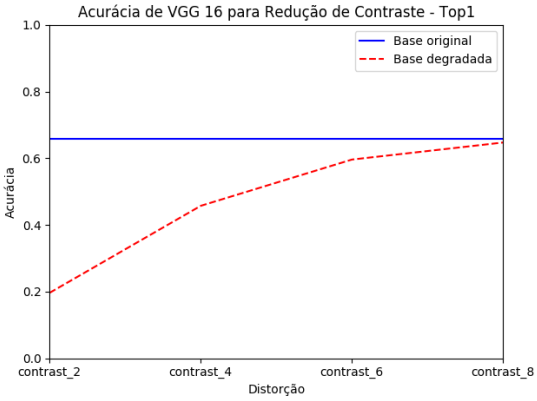
(a) Rede AlexNet



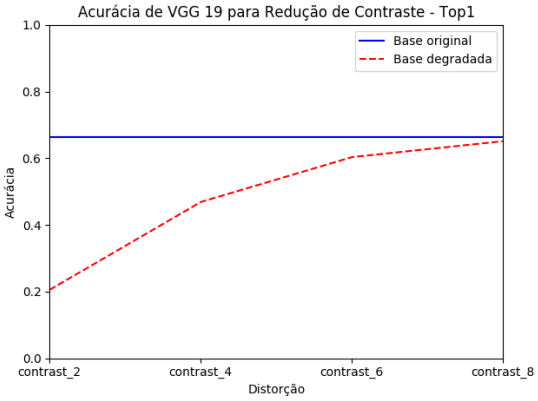
(b) Rede CaffeNet



(c) Rede GoogleNet

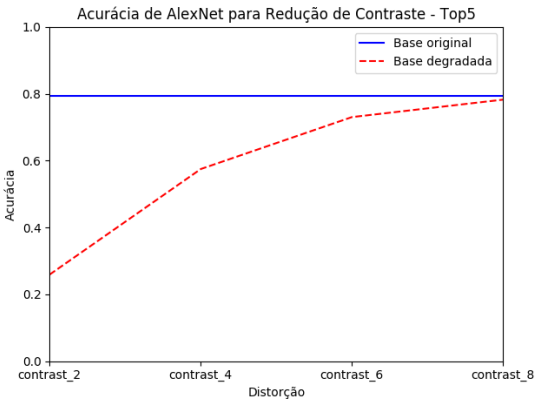


(d) Rede VGG de 16 Camadas

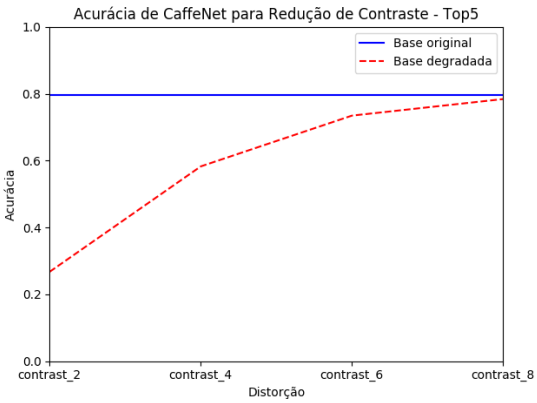


(e) Rede VGG de 19 Camadas

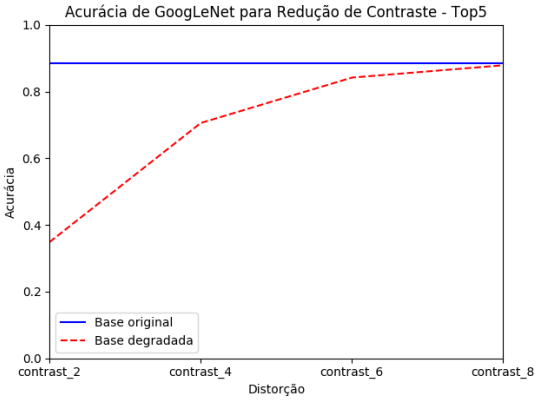
Figura I.5: Acurácia Top1 das redes avaliadas para redução de contraste.



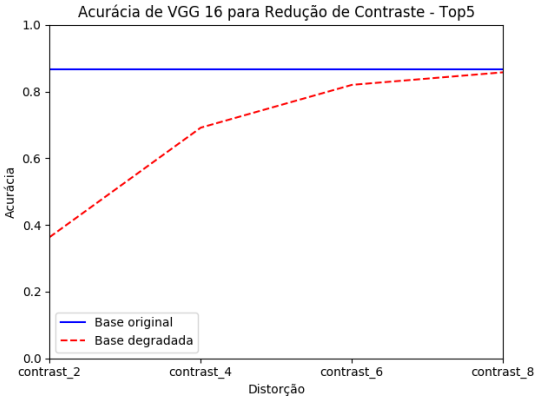
(a) Rede AlexNet



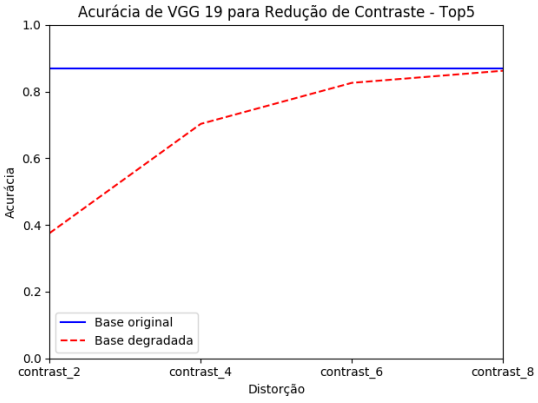
(b) Rede CaffeNet



(c) Rede GoogleNet

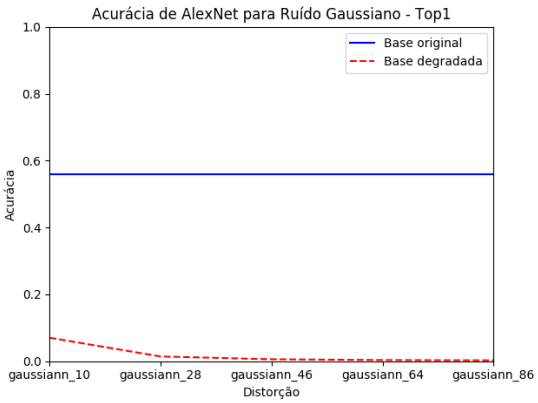


(d) Rede VGG de 16 Camadas

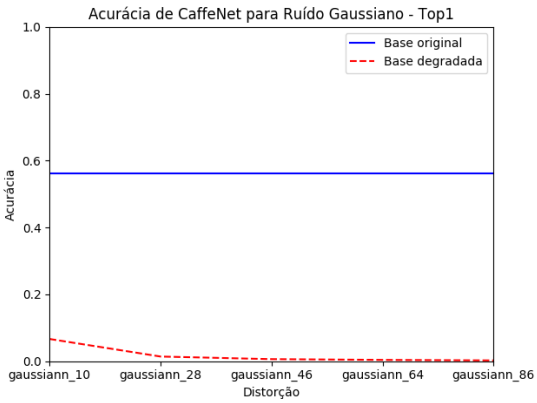


(e) Rede VGG de 19 Camadas

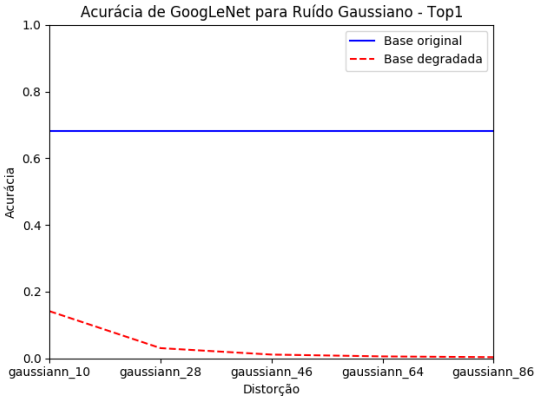
Figura I.6: Acurácia Top5 das redes avaliadas para redução de contraste.



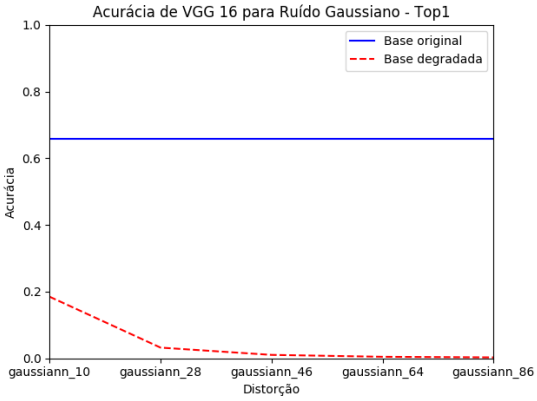
(a) Rede AlexNet



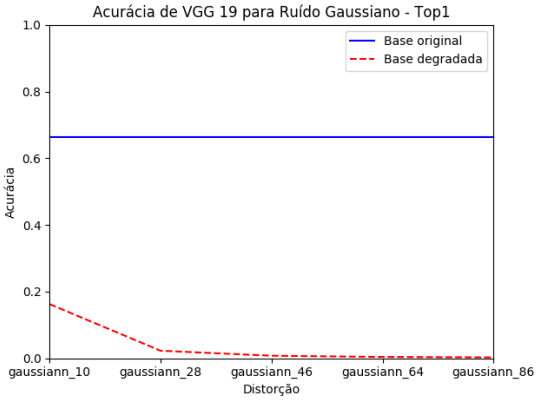
(b) Rede CaffeNet



(c) Rede GoogleNet

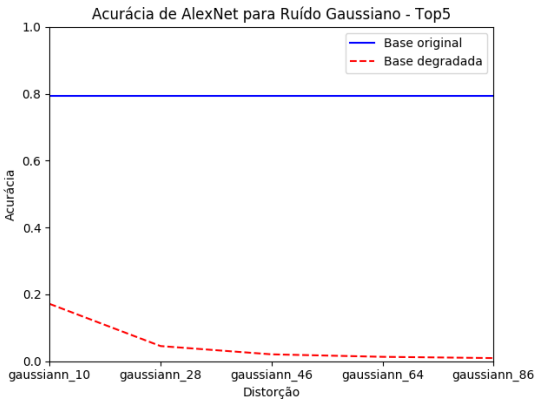


(d) Rede VGG de 16 Camadas

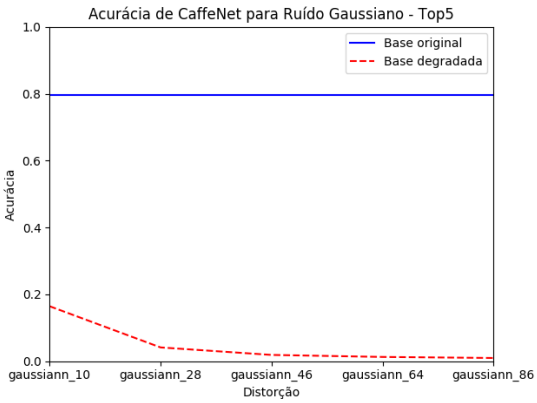


(e) Rede VGG de 19 Camadas

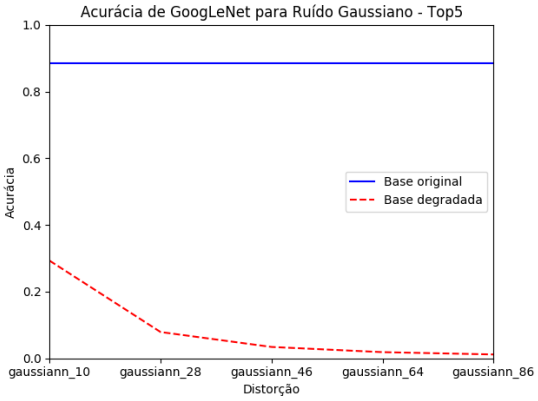
Figura I.7: Acurácia Top1 das redes avaliadas para adição de ruído gaussiano.



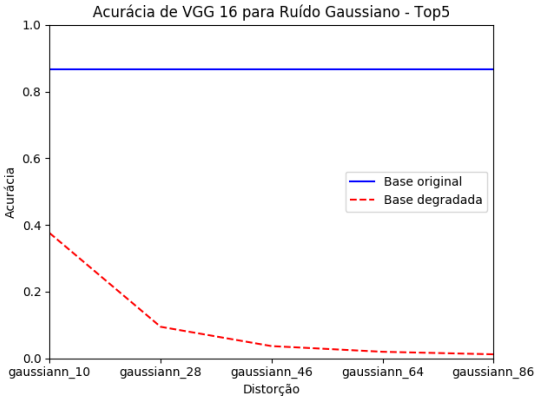
(a) Rede AlexNet



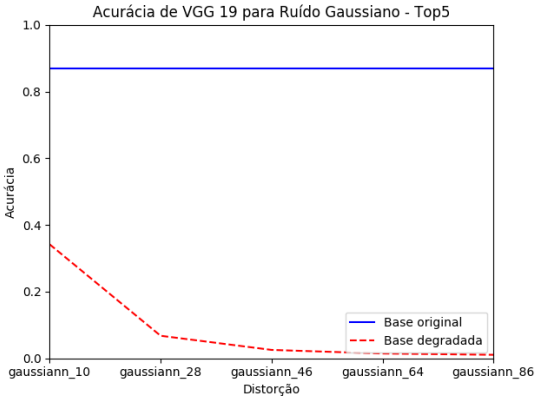
(b) Rede CaffeNet



(c) Rede GoogleNet

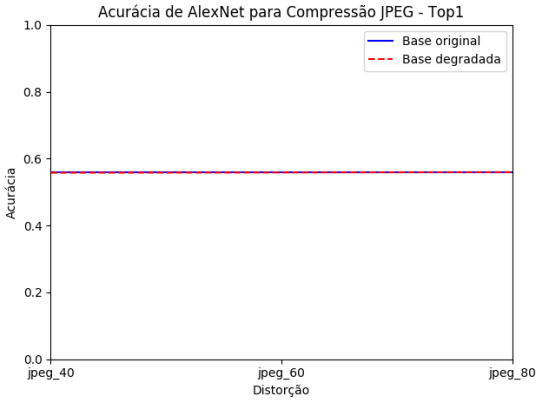


(d) Rede VGG de 16 Camadas

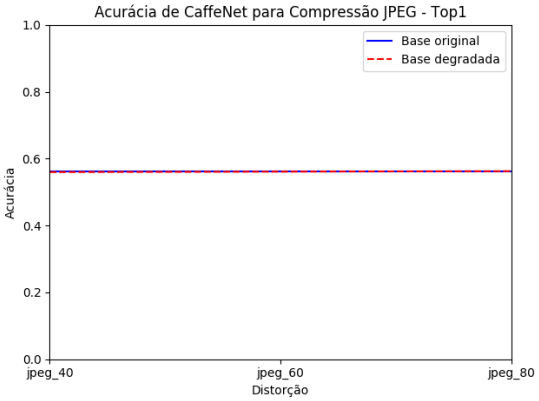


(e) Rede VGG de 19 Camadas

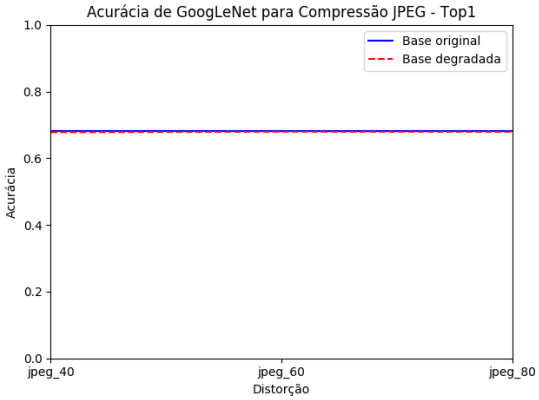
Figura I.8: Acurácia Top5 das redes avaliadas para adição de ruído gaussiano.



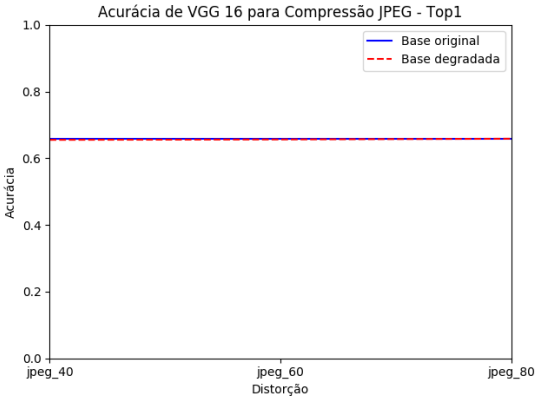
(a) Rede AlexNet



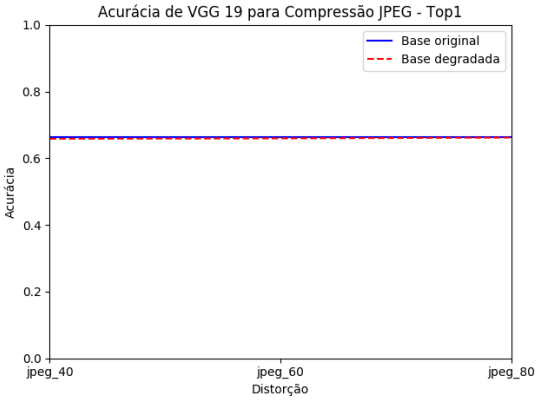
(b) Rede CaffeNet



(c) Rede GoogleNet

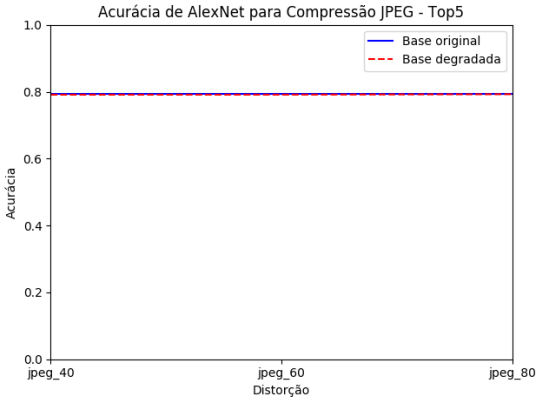


(d) Rede VGG de 16 Camadas

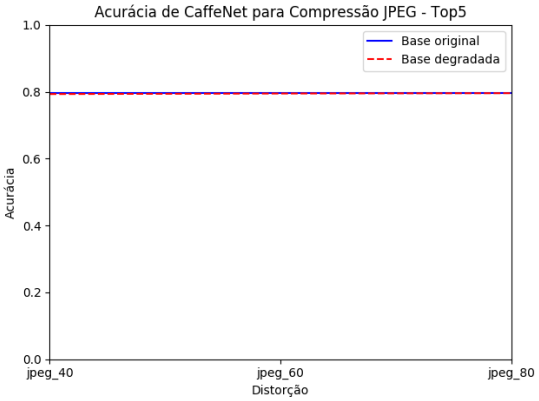


(e) Rede VGG de 19 Camadas

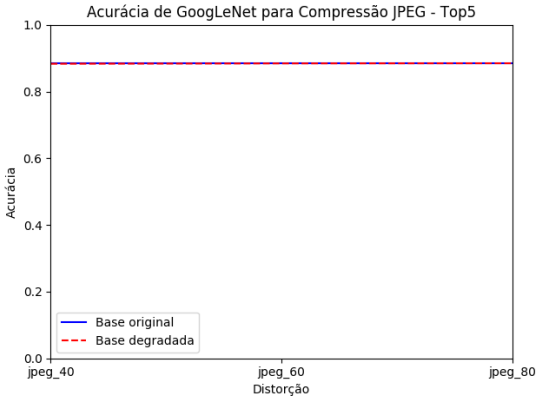
Figura I.9: Acurácia Top1 das redes avaliadas para compressões JPEG.



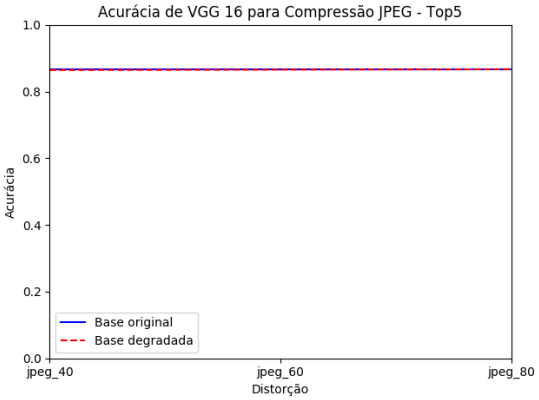
(a) Rede AlexNet



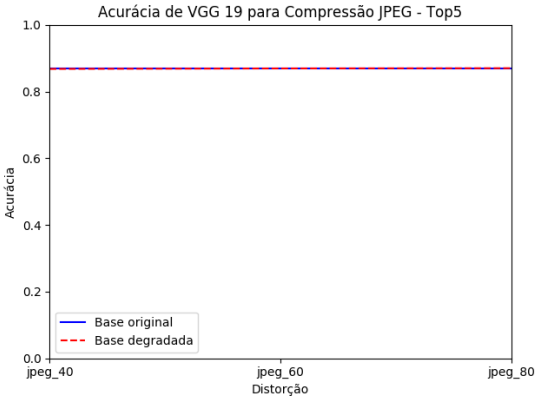
(b) Rede CaffeNet



(c) Rede GoogleNet

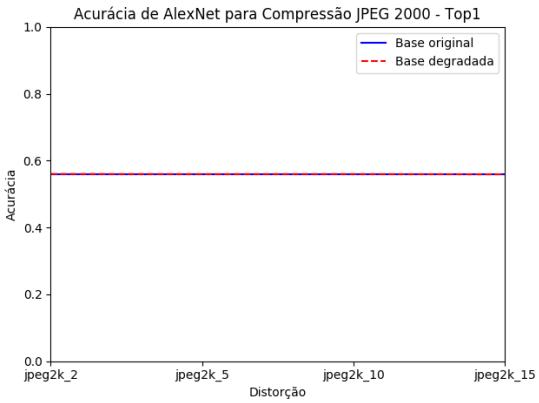


(d) Rede VGG de 16 Camadas

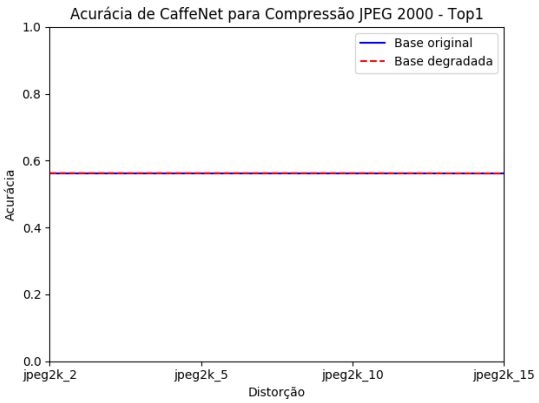


(e) Rede VGG de 19 Camadas

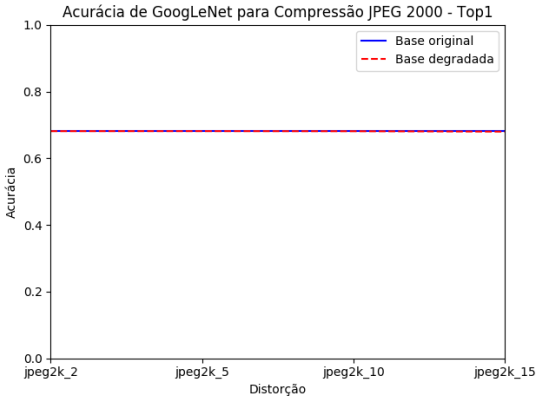
Figura I.10: Acurácia Top5 das redes avaliadas para compressões JPEG.



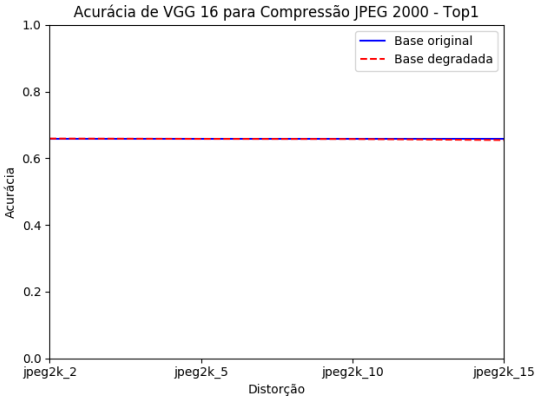
(a) Rede AlexNet



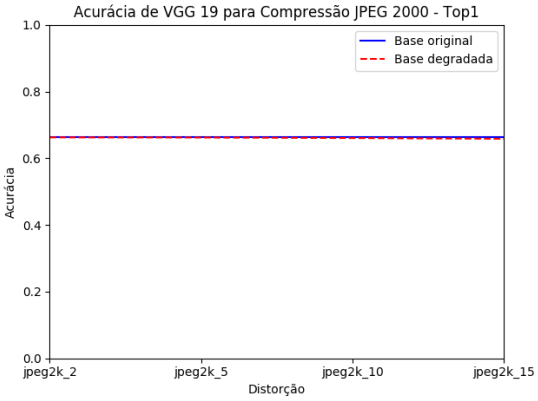
(b) Rede CaffeNet



(c) Rede GoogleNet

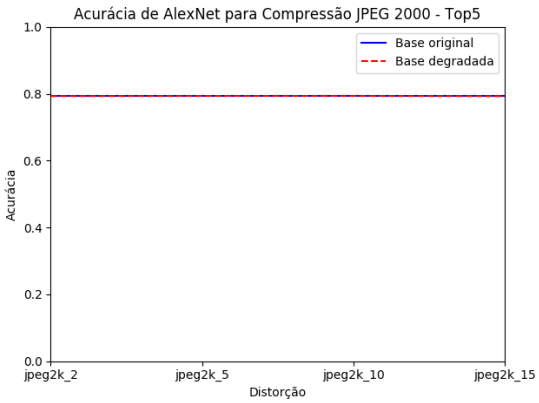


(d) Rede VGG de 16 Camadas

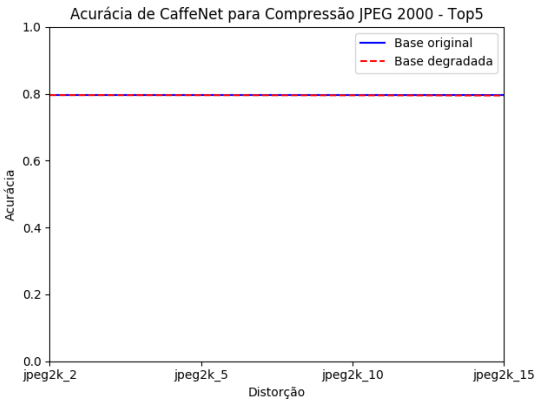


(e) Rede VGG de 19 Camadas

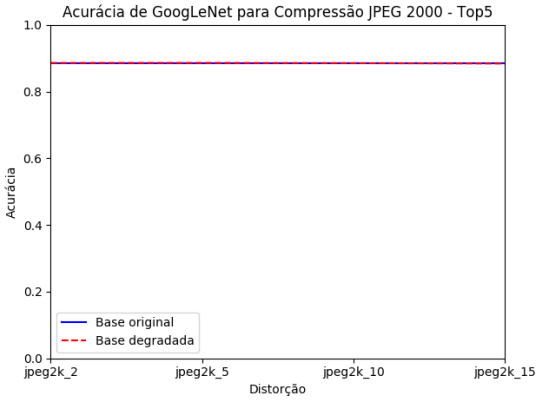
Figura I.11: Acurácia Top1 das redes avaliadas para compressões JPEG2000.



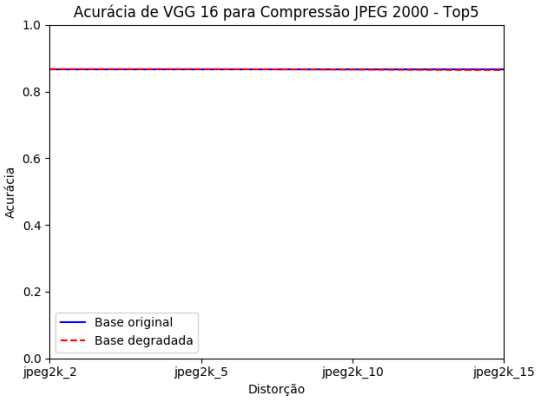
(a) Rede AlexNet



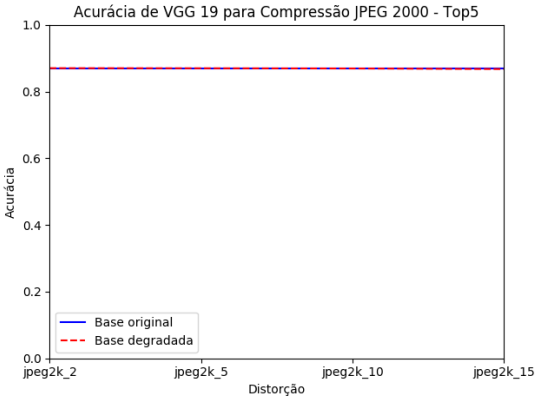
(b) Rede CaffeNet



(c) Rede GoogleNet

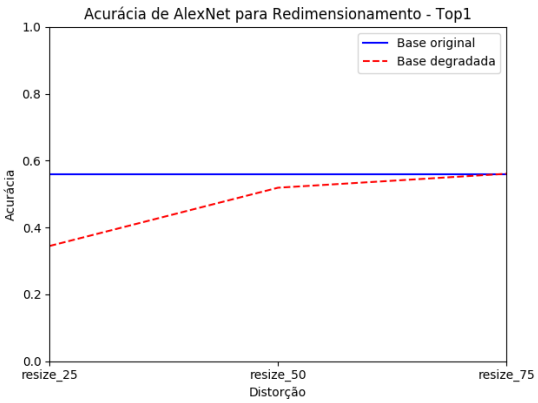


(d) Rede VGG de 16 Camadas

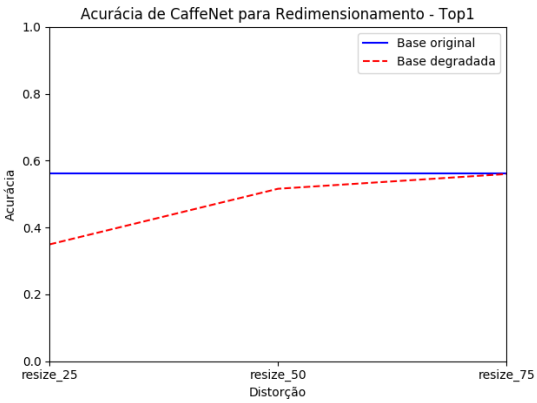


(e) Rede VGG de 19 Camadas

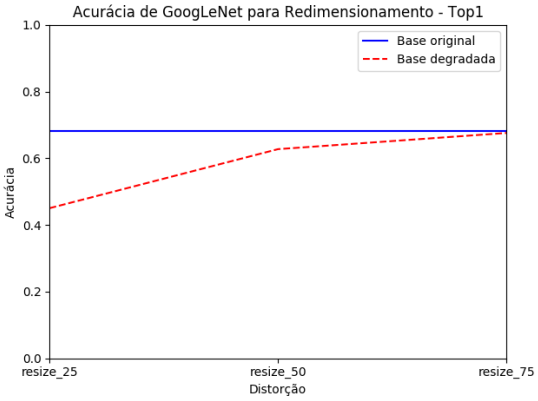
Figura I.12: Acurácia Top5 das redes avaliadas para compressões JPEG2000.



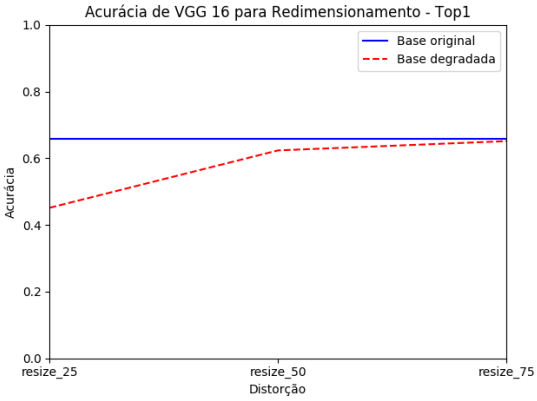
(a) Rede AlexNet



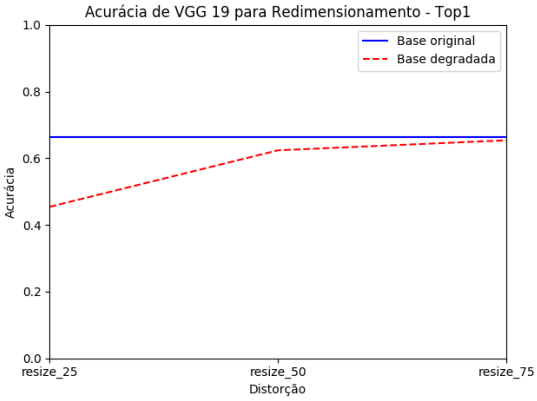
(b) Rede CaffeNet



(c) Rede GoogleNet

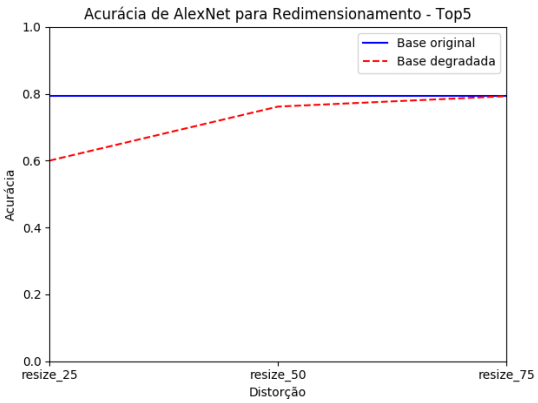


(d) Rede VGG de 16 Camadas

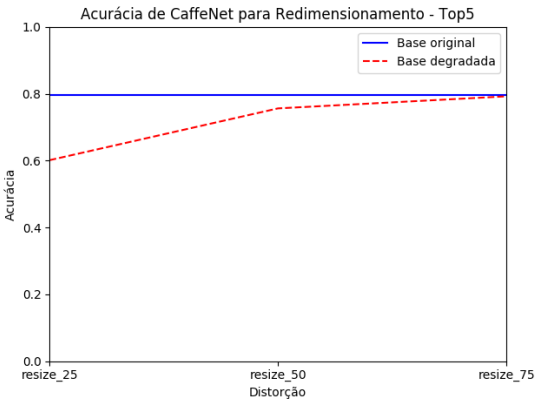


(e) Rede VGG de 19 Camadas

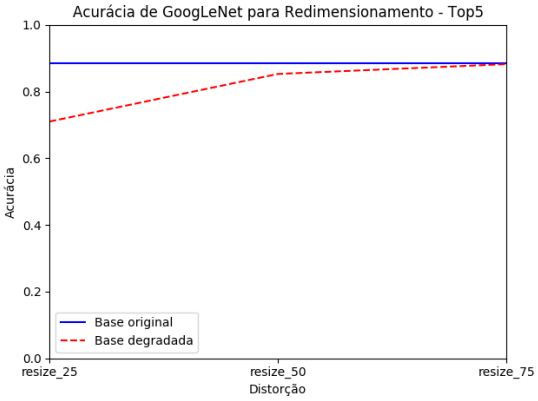
Figura I.13: Acurácia Top1 das redes avaliadas para redimensionamento.



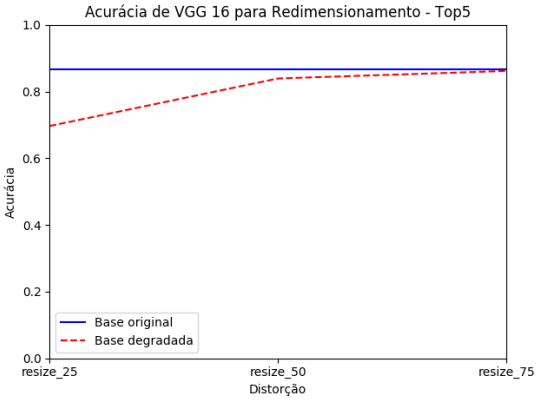
(a) Rede AlexNet



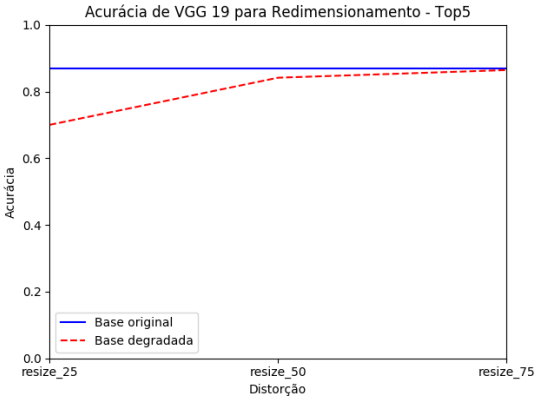
(b) Rede CaffeNet



(c) Rede GoogleNet



(d) Rede VGG de 16 Camadas



(e) Rede VGG de 19 Camadas

Figura I.14: Acurácia Top5 das redes avaliadas para redimensionamento.

II. TABELAS

	Acurácia			
Variante	Top1 Ref.	Top1	Top5 Ref.	Top5
val_blur_3	56,00	53,96	79,34	77,79
val_blur_5	56,00	47,05	79,34	72,12
val_blur_7	56,00	42,56	79,34	67,78
val_blur_9	56,00	37,77	79,34	62,79
val_blur_11	56,00	32,35	79,34	56,26
val_color_8	56,00	41,57	79,34	66,16
val_color_16	56,00	48,89	79,34	73,38
val_color_32	56,00	52,50	79,34	76,36
val_color_64	56,00	54,15	79,34	77,80
val_color_128	56,00	54,92	79,34	78,43
val_contrast_2	56,00	12,45	79,34	25,92
val_contrast_4	56,00	34,62	79,34	57,50
val_contrast_6	56,00	48,73	79,34	73,04
val_contrast_8	56,00	54,67	79,34	78,26
val_gaussiann_10	56,00	7,00	79,34	17,13
val_gaussiann_28	56,00	1,39	79,34	4,51
val_gaussiann_46	56,00	0,56	79,34	2,03
val_gaussiann_64	56,00	0,32	79,34	1,29
val_gaussiann_86	56,00	0,23	79,34	0,91
val_jpeg_40	56,00	55,79	79,34	79,13
val_jpeg_60	56,00	55,86	79,34	79,14
val_jpeg_80	56,00	55,95	79,34	79,27
val_jpeg2k_2	56,01	56,00	79,34	79,32
val_jpeg2k_5	56,01	55,95	79,34	79,36
val_jpeg2k_10	56,01	55,97	79,34	79,38
val_jpeg2k_15	56,01	55,89	79,34	79,30
val_resize_25	56,00	34,46	79,34	60,01
val_resize_50	56,00	51,89	79,34	76,17
val_resize_75	56,00	56,07	79,34	79,29

Tabela II.1: Acurácias calculadas para a rede AlexNet

Variante	Acurácia			
	Top1 Ref.	Top1	Top5 Ref.	Top5
val_blur_3	56,26	54,67	79,58	77,98
val_blur_5	56,26	47,80	79,58	72,61
val_blur_7	56,26	42,88	79,58	68,21
val_blur_9	56,26	38,40	79,58	63,36
val_blur_11	56,26	32,85	79,58	57,03
val_color_8	56,26	41,70	79,58	66,48
val_color_16	56,26	49,19	79,58	73,46
val_color_32	56,26	52,53	79,58	76,70
val_color_64	56,26	54,30	79,58	78,12
val_color_128	56,26	55,17	79,58	78,70
val_contrast_2	56,26	12,96	79,58	26,75
val_contrast_4	56,26	34,88	79,58	58,29
val_contrast_6	56,26	49,34	79,58	73,48
val_contrast_8	56,26	54,84	79,58	78,43
val_gaussiann_10	56,26	6,63	79,58	16,43
val_gaussiann_28	56,26	1,37	79,58	4,10
val_gaussiann_46	56,26	0,59	79,58	1,86
val_gaussiann_64	56,26	0,37	79,58	1,25
val_gaussiann_86	56,26	0,21	79,58	0,94
val_jpeg_40	56,26	55,97	79,58	79,32
val_jpeg_60	56,26	56,11	79,58	79,48
val_jpeg_80	56,26	56,25	79,58	79,58
val_jpeg2k_2	56,26	56,26	79,58	79,60
val_jpeg2k_5	56,26	56,24	79,58	79,58
val_jpeg2k_10	56,26	56,24	79,58	79,54
val_jpeg2k_15	56,26	56,15	79,58	79,48
val_resize_25	56,26	34,93	79,58	60,14
val_resize_50	56,26	51,58	79,58	75,63
val_resize_75	56,26	55,99	79,58	79,24

Tabela II.2: Acurácias calculadas para a rede CaffeNet

Variante	Acurácia			
	Top1 Ref.	Top1	Top5 Ref.	Top5
val_blur_3	68,16	65,75	88,57	87,00
val_blur_5	68,16	60,47	88,57	83,65
val_blur_7	68,16	55,62	88,57	79,98
val_blur_9	68,16	50,75	88,57	75,70
val_blur_11	68,16	45,05	88,57	70,26
val_color_8	68,16	52,75	88,57	77,11
val_color_16	68,16	61,19	88,57	83,78
val_color_32	68,16	64,74	88,57	86,38
val_color_64	68,16	66,33	88,57	87,50
val_color_128	68,16	67,15	88,57	87,97
val_contrast_2	68,16	18,99	88,57	34,87
val_contrast_4	68,16	46,93	88,57	70,58
val_contrast_6	68,16	61,88	88,57	84,19
val_contrast_8	68,16	66,97	88,57	87,89
val_gaussiann_10	68,16	14,11	88,57	29,26
val_gaussiann_28	68,16	3,03	88,57	7,86
val_gaussiann_46	68,16	1,10	88,57	3,39
val_gaussiann_64	68,16	0,56	88,57	1,83
val_gaussiann_86	68,16	0,33	88,57	1,14
val_jpeg_40	68,16	67,79	88,57	88,33
val_jpeg_60	68,16	67,96	88,57	88,44
val_jpeg_80	68,16	67,93	88,57	88,48
val_jpeg2k_2	68,15	68,12	88,57	88,58
val_jpeg2k_5	68,15	68,13	88,57	88,60
val_jpeg2k_10	68,15	68,10	88,57	88,51
val_jpeg2k_15	68,15	67,98	88,57	88,40
val_resize_25	68,16	45,02	88,57	71,01
val_resize_50	68,16	62,72	88,57	85,28
val_resize_75	68,16	67,58	88,57	88,23

Tabela II.3: Acurácias calculadas para a rede GoogleNet

Variante	Acurácia			
	Top1 Ref.	Top1	Top5 Ref.	Top5
val_blur_3	65,90	63,47	86,73	84,86
val_blur_5	65,90	58,51	86,73	80,99
val_blur_7	65,90	52,78	86,73	76,27
val_blur_9	65,90	47,18	86,73	71,30
val_blur_11	65,90	41,86	86,73	65,99
val_color_8	65,90	49,64	86,73	73,91
val_color_16	65,90	58,46	86,73	81,44
val_color_32	65,90	62,14	86,73	84,16
val_color_64	65,90	64,07	86,73	85,39
val_color_128	65,90	64,86	86,73	85,99
val_contrast_2	65,90	19,65	86,73	36,35
val_contrast_4	65,90	45,75	86,73	69,18
val_contrast_6	65,90	59,62	86,73	82,01
val_contrast_8	65,90	64,73	86,73	85,78
val_gaussiann_10	65,90	18,46	86,73	37,56
val_gaussiann_28	65,90	3,19	86,73	9,44
val_gaussiann_46	65,90	1,01	86,73	3,64
val_gaussiann_64	65,90	0,44	86,73	1,95
val_gaussiann_86	65,90	0,25	86,73	1,18
val_jpeg_40	65,90	65,51	86,73	86,43
val_jpeg_60	65,90	65,63	86,73	86,56
val_jpeg_80	65,90	65,85	86,73	86,69
val_jpeg2k_2	65,89	65,90	86,73	86,73
val_jpeg2k_5	65,89	65,81	86,73	86,72
val_jpeg2k_10	65,89	65,77	86,73	86,58
val_jpeg2k_15	65,89	65,46	86,73	86,44
val_resize_25	65,90	45,13	86,73	69,66
val_resize_50	65,90	62,35	86,73	83,92
val_resize_75	65,90	65,13	86,73	86,18

Tabela II.4: Acurácias calculadas para a rede VGG16

Variante	Acurácia			
	Top1 Ref.	Top1	Top5 Ref.	Top5
val_blur_3	66,27	63,61	87,03	85,07
val_blur_5	66,27	58,59	87,03	81,37
val_blur_7	66,27	53,37	87,03	76,74
val_blur_9	66,27	48,18	87,03	72,12
val_blur_11	66,27	43,04	87,03	67,17
val_color_8	66,27	50,90	87,03	75,36
val_color_16	66,27	59,41	87,03	82,04
val_color_32	66,27	63,02	87,03	84,63
val_color_64	66,27	64,58	87,03	85,70
val_color_128	66,27	65,26	87,03	86,35
val_contrast_2	66,27	20,55	87,03	37,56
val_contrast_4	66,27	46,89	87,03	70,31
val_contrast_6	66,27	60,34	87,03	82,62
val_contrast_8	66,27	65,12	87,03	86,25
val_gaussiann_10	66,27	16,24	87,03	34,19
val_gaussiann_28	66,27	2,26	87,03	6,75
val_gaussiann_46	66,27	0,75	87,03	2,50
val_gaussiann_64	66,27	0,39	87,03	1,42
val_gaussiann_86	66,27	0,26	87,03	1,04
val_jpeg_40	66,27	65,84	87,03	86,78
val_jpeg_60	66,27	65,96	87,03	86,93
val_jpeg_80	66,27	66,20	87,03	87,00
val_jpeg2k_2	66,26	66,26	87,02	87,01
val_jpeg2k_5	66,26	66,22	87,02	86,98
val_jpeg2k_10	66,26	66,06	87,02	86,92
val_jpeg2k_15	66,26	65,83	87,02	86,74
val_resize_25	66,27	45,42	87,03	70,01
val_resize_50	66,27	62,38	87,03	84,15
val_resize_75	66,27	65,38	87,03	86,44

Tabela II.5: Acurácias calculadas para a rede VGG19